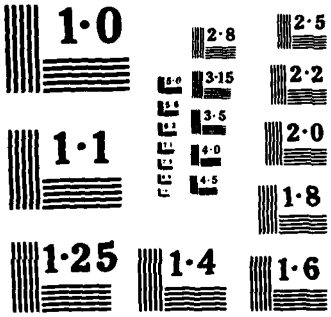
AD-A162 843 COMMUNICATION AND MISCOMMUNICATION(U) BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA B A GOODMAN OCT 85 BBN-5681 N00014-85-C-0079 1/3

UNCLASSIFIED F/G 9/2 NL


1·0
2·8
2·5
3·15
2·2
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

BBN Laboratories Incorporated

A Subsidiary of Bolt Beranek and Newman Inc.

6

Report No. 5681

Communication and Miscommunication

October 1985

Prepared for:
Defense Advanced Research Projects Agency

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER BBN Report No. 5681	2. GOVT ACCESSION NO. AD-A162843	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMMUNICATION AND MISCOMMUNICATION		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER BBN Report No. 5681
7. AUTHOR(s) Bradley Alan Goodman		8. CONTRACT OR GRANT NUMBER(s) N00014-77-C-0378 N00014-85-C-0079
9. PERFORMING ORGANIZATION NAME AND ADDRESS BBN Laboratories Inc. 10 Moulton St. Cambridge, MA 02238		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy Arlington, VA 22217		12. REPORT DATE October 1985
		13. NUMBER OF PAGES 220
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial intelligence, computational linguistics, natural language understanding, knowledge representation, miscommunication, discourse, KL-ONE.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report discusses one aspect of enabling people to communicate in natural language with computers. The central focus of this work is a study on how one could build robust natural language processing systems that can detect and recover from miscommunication. The study of miscommunication is a necessary task within such a context since any computer capable of communicating with humans in natural language must be tolerant of the complex, imprecise, or ill-devised utterances that people often

cont'd

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract (cont'd.)

use. This goal first required an inquiry into how people communicate and how they recover from problems in communication. That investigation centered on the kinds of miscommunication that occur in human communication with a special emphasis on reference problems, i.e., problems a listener has determining whom or what a speaker is talking about. A collection of protocols of a speaker explaining to a listener how to assemble a toy water pump were studied and the common errors seen in speakers' descriptions were categorized. This ~~study~~ led to the development of techniques for avoiding failures of reference that were employed in the reference identification component of a natural language understanding program.

The traditional approaches to reference identification in previous natural language systems were found to be less elaborate than people's real behavior. In particular, listeners often find the correct referent even when the speaker's description does not describe any object in the world. To model a listener's behavior, a new component was added to the traditional reference identification mechanism to resolve difficulties in a speaker's description. This new component uses knowledge about linguistic and physical context in a negotiation process that determines the most likely places for error in the speaker's utterance. The actual repair of the speaker's description is achieved by using the knowledge sources to guide relaxation techniques that delete or replace portions of the description. The algorithm developed more closely approximates people's behavior.

Report No. 5681

COMMUNICATION AND MISCOMMUNICATION

Bradley Alan Goodman

October 1985

Prepared for:

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

This report is a revised version of a thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign, 1984.

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-77-C-0378 and N00014-85-C-0079. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

To My Mother

In Loving Memory of

My Father

And to My Aunt

and Uncle

ACKNOWLEDGEMENTS

I want to thank the many, many people who have contributed to this work, and provided support and encouragement to me.

To Candy Sidner, my advisor at BBN, for taking such a strong interest in my work, and for her insightful comments and suggestions during the course of this work. Candy encouraged me to pursue my thesis work and then provided much guidance, support and enthusiasm.

To Dave Waltz, my thesis advisor, for his ideas, support and encouragement throughout my graduate career. Dave provided me with many tools and opportunities to do good research. Even when one of those opportunities meant leaving Illinois, he unselfishly allowed me to pursue my thesis work at BBN.

To Phil Cohen, for urging me to pursue my thesis topic. Phil's advice at the critical initial stage of my work really got me moving.

To Bill Woods, for fostering such a great environment to perform research, and for his suggestions and support during the course of this work.

To Walter Reitman, for his support and encouragement and for taking the helm and making 1984 run smoother than any of us could have ever expected.

To my sisters, my brother, and my mother, for giving everything and more. My family was always there providing me with much love, encouragement and support. They kindly overlooked the five years in a row that I told them I was "one year" from finishing.

To my Aunt, Uncle, and cousins, for providing me with so much love, encouragement and support.

To Jeff Gibbons, for his close friendship, encouragement, guidance, and all those hours of stimulating technical discussions. Jeff always provided a wealth of good ideas whenever I got stuck.

To Kathy Larkin, for her friendship, wit, and for providing me with the job that allowed me to stay at BBN to pursue my thesis work. Working with Kathy on "Project X" was one of my most memorable times at BBN.

To Marc Vilain, for his warm friendship, encouragement and numerous discussions about my work. Marc read many drafts of my papers and his comments made this document so much more readable.

To Robyn Wolff, for her faith in me and for making the last part of my graduate career so much more enjoyable.

To all the members of the Advanced Automation Research Group at the Coordinated Science Laboratory, for providing such a great place for me to grow. Special thanks to Tim Finin, George Hadden and Harry Tennant for the many, many discussions we had over the years and for all their friendship and encouragement.

To all the people at BBN over the years, for providing such a great research environment. Special thanks to Phil Cohen, Scott Fertig and Kathy Starr for providing me with their water pump dialogues and for their invaluable observations on them. And to Diane Litman, Jim Schmolze, and Bonnie Webber for all their discussions, comments and encouragement during this work. Thanks also to Peggy Moser for getting IRUS to parse and interpret sample utterances in the water pump domain.

To my friends and colleagues, many mentioned above, and especially to Lyn Bates, Rusty Bobrow, Ron Brachman, Janet Finin, Howard Finkelstein, Andy Haas, Bob Ingria, David Israel, Miriam Kurland, Maureen Saffi, Dave Stallard, Mira Sussman, Beverly Tobiason, Bonnie Waltz, and Suzie Weaver.

This research was supported in part by the Defense Advanced Research Project Agency under contract N00014-77-C-0378 and N00014-85-C-0079.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Communication and miscommunication

My goal is to build robust natural language processing systems that can detect and recover from miscommunication. The development of such systems requires a study on how people communicate and how they recover miscommunication. This thesis is an investigation of the kinds of miscommunication that occur in human communication with a special emphasis on reference problems, i.e., problems a listener has determining whom or what a speaker is talking about. I have written computer programs and algorithms that demonstrate how one could solve such problems in a natural language understanding system. The study of miscommunication is a necessary task for natural language understanding systems since any computer capable of communicating with humans in natural language must be tolerant of the complex, imprecise, or ill-devised utterances that people often use.

Communication involves a series of utterances from a speaker to a hearer. The hearer uses these utterances to access his own knowledge and the world around him. Some of these utterances are noun phrases that refer to objects, places, ideas and people that exist in the real world or in some imaginary world. They cannot be considered in isolation. For example, consider the utterance "Give me that thing." It can be uttered in many different situations and can result in different referents of "that thing." Understanding such referring expressions requires the hearer to take into account the speaker's intention, the speaker's overall goal, the beliefs of the speaker and hearer, the linguistic context, the physical context, and the syntax and semantics of the current utterance. The hearer could misinterpret the speaker's information in any one of these parts of communication. Such misunderstandings constitute miscommunication. In this research I focused primarily on effects of the linguistic context and the physical context.

To explore such reference problems, the following method was devised and followed. First, I analyzed protocols of subjects communicating about a task. I isolated knowledge that people have about the world and about language that is used to recover from reference miscommunications. I designed algorithms to apply a person's knowledge about linguistic and physical context to determine the most likely places for error in the speaker's utterance. I then wrote computer programs to represent a spatially complex physical world, to manipulate the structure of that representation to reflect the changes caused by the listener's interpretation of the speaker's utterances and physical actions to the world, to perform referent identification on noun phrases, and, when referent identification failed, to search the physical world for reasonable candidates for the referent. These programs and their underlying algorithms form one component of a natural language system.

One goal in the rest of this chapter is to illustrate how my current views on reference identification depart from views held by other researchers in artificial intelligence. Another goal is to show where this research fits in the scheme of natural language understanding by computers. Finally, the chapter summarizes the approach of this research.

1.2 A new reference paradigm from a computational viewpoint

Reference identification is a search process where a listener looks for something in the world that satisfies a speaker's uttered description. A computational scheme for performing such reference identifications has evolved from work by other artificial intelligence researchers (e.g., see [30], [37] and the discussion in Chapter 3). That traditional approach succeeds if a referent is found, or fails if no referent is found (see Figure 1-1(a)). However, a reference identification component must be more versatile than those previously constructed. The excerpts provided in Chapter 2 will show that the traditional approach is inadequate because people's real behavior is much more elaborate. In particular, listeners often find the correct referent even when the speaker's description does not describe any object in the world. For example, a speaker could describe a turquoise block as the "blue block." Most listeners would go ahead and assume that the turquoise block was the one the speaker meant since turquoise and blue are similar colors.

A key feature to reference identification is "negotiation." Negotiation in reference identification comes in two forms. First, it can occur between the listener and the speaker. The listener can step back, expand greatly on the speaker's description of a plausible referent, and ask for confirmation that he has indeed found the correct referent. For example, a listener could initiate negotiation with "I'm confused. Are you talking about the thing that is kind of flared at the top? Couple inches long. It's kind of blue." Second, negotiation can be with oneself. This self-negotiation is the one that I am most concerned with in this research. The listener considers aspects of the speaker's description, the context of the communication, the listener's own abilities, and other relevant sources of knowledge. He then applies that deliberation to determine whether one referent candidate is better than another or, if no candidate is found, what are the most likely places for error or confusion. Such negotiation can result in the listener testing whether or not a particular referent works. For example, linguistic descriptions can influence a listener's perception of the world. The listener must ask himself whether he can perceive one of the objects in the world the way the speaker described it. In some cases, the listener's perception may overrule parts of the description because the listener can't perceive it the way the speaker described it.

To repair the traditional approach I have developed an algorithm that captures for certain cases the listener's ability to negotiate with himself for a referent. It can search for a referent and, if it doesn't find one, it can try to find possible referent candidates that might work, and then loosen the speaker's description using knowledge about the speaker, the conversation, anᵈ the listener himself. Thus, the reference process becomes multi-step and resumable. This computational model, which I call "FWIM" for "Find What I Mean", is more faithful to the data than the traditional model (see Figure 1-1(b)).

One means of making sense of a failed description is to delete or replace portions of it that cause it not to match objects in the hearer's world. In my program I am using "relaxation" techniques to capture this behavior. My reference identification module treats descriptions as approximate. It relaxes a description in order to find a referent when the literal content of the description fails to provide the needed information. Relaxation, however, is not performed blindly on the description. I try to model a person's behavior by drawing on sources of knowledge used by people. I have developed a computational model that can relax aspects of a description using many of these sources of knowledge. Relaxation then becomes a form of communication repair (in the style of the work on repair theory developed in [11]).

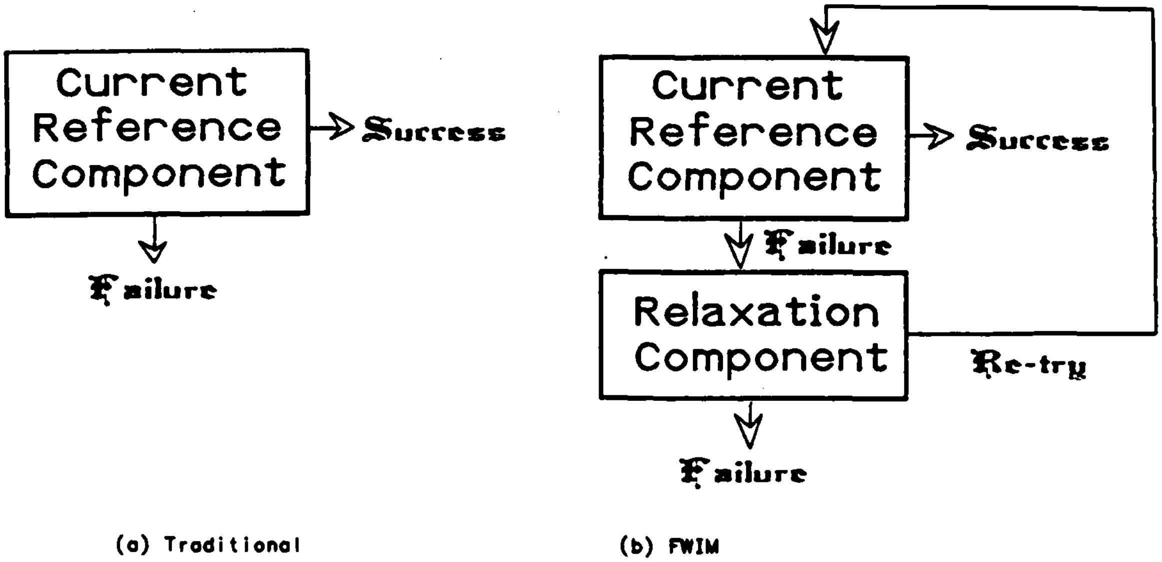


Figure 1-1: Approaches to *reference identification*

The relaxation component of the reference identification module is described in Chapters 5 and 6.

1.3 Context of the research

This section introduces the structure of the BBN natural language system currently under development, and points out why it and systems like it need to handle miscommunication. It also describes the particular domain which was studied to motivate many of the results in this work and that was simulated in the computer programs.

1.3.1 The BBN natural language system

The work described here is part of a larger effort [71, 72] to build a natural language understanding system. The system is organized as shown in Figure 1-2. For our purposes, a "speaker" types input in English using a terminal. The speaker's input is analyzed by the parser. The parser consults a grammar and a dictionary and passes parsed constituents on to the semantic interpreter. The semantic interpreter can accept or reject the parse on semantic grounds. Once a version of the parse is

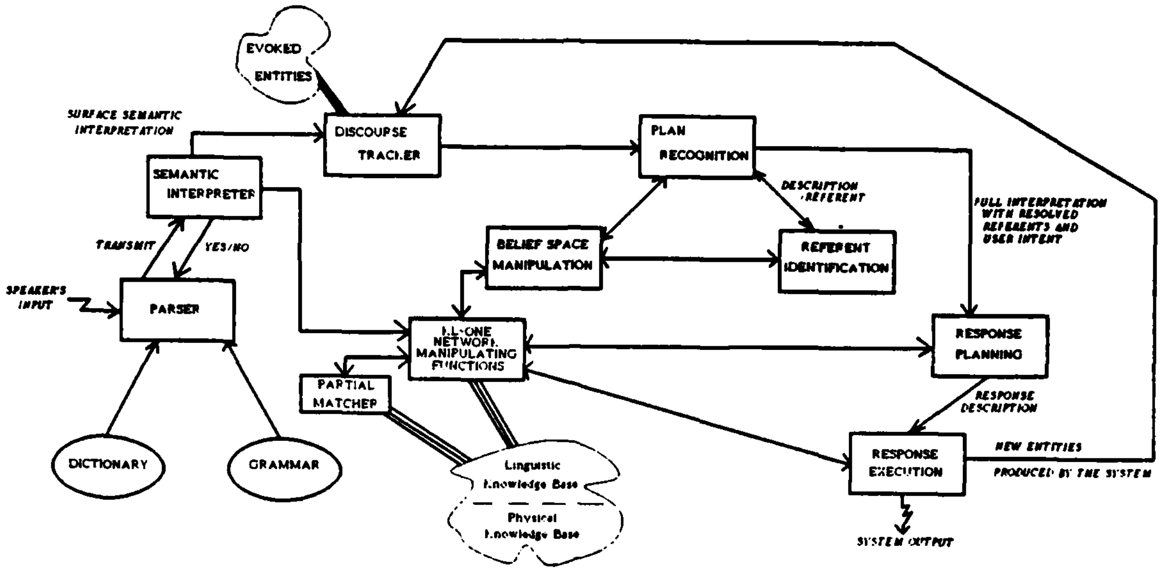


Figure 1-2: System structure

accepted, the semantic interpretation of the speaker's input is passed to the discourse tracker. The discourse tracker follows the relevant elements under discussion in the conversation, noting if shifts are made from the current elements to new ones. It passes the interpretation to the plan recognition module. This module must determine what the speaker wants the system to do, i.e., discover what is the goal of the speaker, how that goal fits into plans available to the system for achieving goals, which particular plan should be used, and how to fill in that plan with information provided in the speaker's input. The plan recognizer consults with the belief space manipulation and referent identification modules. The belief space module can manipulate representations of the system's beliefs about the speaker and about the system's capabilities. This allows the system to make inferences beyond the literal content of the speaker's input, getting to the speaker's intent. The referent identifier is called by the plan recognition module to find referents for entities in a plan. It takes descriptions from the speaker's input and returns a pointer to the actual entities (or some representation thereof) described in the input. From this information, the plan recognizer passes a complete interpretation of the speaker's request to the response planner. The response planner's task is to determine how to respond to the speaker's request. Once such a plan is formulated, it is passed to the response execution module which executes the plan.

This system is currently under construction though major components have been completed. The parser, semantic interpreter, KL-One functions, reference identification component and partial matcher have been implemented. The extension to the reference identification component to allow relaxation is completely designed and has been partially implemented. The plan recognition module is partially designed and implemented. The rest of the system is currently being designed.

1.3.2 Places where miscommunication occurs

The current research of my colleagues and myself views most dialogues as being cooperative and goal directed, i.e., a speaker and listener work together to achieve a common goal. The interpretation of an utterance involves identifying the underlying plan or goal that the utterance reflects [18, 3, 68, 74]. This plan, however, is rarely, if ever, obvious at the surface sentence level. A central issue in the interpretation of utterances is the transformation of sequences of complex, imprecise, or ill-devised utterances into well-specified plans that might be carried out by dialogue participants. Within this context, miscommunication can occur.

I am particularly concerned with cases of miscommunication from the hearer's viewpoint, such as when the hearer is inattentive to, confused about, or misled about the intentions of the speaker. In ordinary exchanges speakers usually make assumptions regarding what their listeners know about a topic of discussion. They will leave out details thought to be superfluous [5, 49]. Since the speaker really does not know exactly what a listener knows about a topic, it is easy to make statements that can be misinterpreted or not understood by the listener because not enough details were presented. Some of the problems that could be encountered by the listener during interpretation of an utterance include incorrectly identifying the action requested by the speaker and misinterpreting the beliefs and context of the speaker. Another principal source of trouble is the descriptions constructed by the speaker to refer to actual objects in the world. A description can be imprecise, confused, ambiguous or overly specific. It might be interpreted under the wrong context. As a result, reference identification errors occur (I will call these errors "misreference."). The listener cannot determine what object is being described.

Such utterances and descriptions constitute a kind of "ill-formed" input (see

[79] for a discussion on ill-formed input). The blame for ill-formedness may lie partly with the speaker and partly with the listener. The speaker may have been sloppy or not taken the hearer into consideration; the listener may be either remiss or unwilling to admit he can't understand the speaker and to ask the speaker for clarification, or may simply believe that he has understood when he in fact has not.

I have tried to motivate in this section that the natural language paradigm followed by my colleagues, myself, and other researchers leaves plenty of room for miscommunication to occur. Such miscommunication leads to problems for a human listener and should, thus, cause similar problems in a natural language understanding program. This work is meant to be part of an on-going effort to develop a reference identification and plan recognition mechanism that can exhibit more "human-like" tolerance of such ill-formed utterances.

1.3.3 Kinds of dialogue studied

I am following the task-oriented paradigm of Grosz [30] since it is easy to study (through videotapes), it places the world in front of you (a primarily extensional world), and it limits the discussion while still providing a rich environment for complex descriptions. The task chosen as the target for the system is the assembly of a toy water pump. The water pump is reasonably complex, containing four subassemblies that are built from plastic tubes, nozzles, valves, plungers, and caps that can be screwed or pushed together. A large corpus of dialogues concerning this task was collected by Cohen (see [20, 21, 22]). These dialogues contained instructions from an "expert" to an "apprentice" that explain the assembly of the pump. Both participants were trying to achieve a common goal - the successful assembly of the pump. This domain is rich in perceptual information, allowing for complex descriptions of elements in it. The data provide examples of imprecision, confusion, and ambiguity as well as attempts to correct these problems.

The following exchange exemplifies one such situation. In it, A is instructing J to assemble part of the water pump. Refer to Figure 1-3(a) for a picture of the pump. A and J are communicating verbally but neither can see the other. (The bracketed text in the excerpt tells what was actually occurring while each utterance was spoken.) Notice the complexity of the speaker's descriptions and the resultant processing required by the listener. This dialogue illustrates that (1) listeners repair the speaker's description in order to find a referent, (2) they repair their initial reference choice once they are given more information, and (3) they can fail to choose a proper referent. In Line 7, A describes the two holes on the *BASEVALVE* as "the little hole." J must repair the description, realizing that A doesn't really mean "one" hole but is referring to the "two" holes. J apparently does this since he doesn't complain about A's description and correctly attaches the *BASEVALVE* to the *TUBEBASE*. Figure 1-3(b) shows the configuration of the pump after the *TUBEBASE* is attached to the *MAINTUBE* in Line 10. In Line 13, J interprets "a red plastic piece" to refer to the *NOZZLE*. When A adds the relative clause "that has four gizmos on it," J is forced to drop the *NOZZLE* as the referent and to select the *SLIDEVALVE*. In Lines 17 and 18, A's description "the other--the open part of the main tube, the lower valve" is ambiguous, and J selects the wrong site, namely the *TUBEBASE*, in which to insert the *SLIDEVALVE*. Since the *SLIDEVALVE* fits, J doesn't detect any trouble. Lines 20 and 21 keep J from thinking that something is wrong because the part fits loosely. In Lines 27 and 28, J indicates that A has not given him enough information to perform the requested action. In Line 30, J further compounds the error in Line 18 by putting the *SPOUT* on the *TUBEBASE*.

Excerpt 1 (Telephone)

A: 1. Now there's a blue cap
[J grabs the TUBEBASE]
2. that has two little teeth sticking
3. out of the bottom of it.

J: 4. Yeah.

A: 5. Okay. On that take the
6. bright shocking pink piece of plastic
[J puts down MAINTUBE and takes BASEVALVE]
7. and stick the little hole over the teeth.
[J starts to install the BASEVALVE, backs off, looks at it again and then goes ahead and installs it]

J: 8. Okay.

A: 9. Now screw that blue cap onto

10. the bottom of the main tube.
[J screws TUBEBASE onto MAINTUBE]

J: 11. Okay.

A: 12. Now, there's a--
13. a red plastic piece
[J starts for NOZZLE]
14. that has four gizmos on it.
[J switches to SLIDEVALVE]

J: 15. Yes.

A: 16. Okay. Put the ungizmoed end in the uh

17. the other--the open

18. part of the main tube, the lower valve.
[J puts SLIDEVALVE into hole in TUBEBASE, but A meant OUTLET2 of MAINTUBE]

J: 19. All right.

A: 20. It just fits loosely. It doesn't
21. have to fit right. Okay, then take
22. the clear plastic elbow joint.
[J takes SPOUT]

J: 23. All right.

A: 24. And put it over the botto:n opening, too.
[J tries installing SPOUT on TUBEBASE]

J: 25. Okay.

A: 26. Okay. Now, take the--

J: 27. Which end am I supposed to put it over?
28. Do you know?

A: 29. Put the--put the--the big end--

30. the big end over it.
[J pushes big end of SPOUT on TUBEBASE, twisting it to force it on]

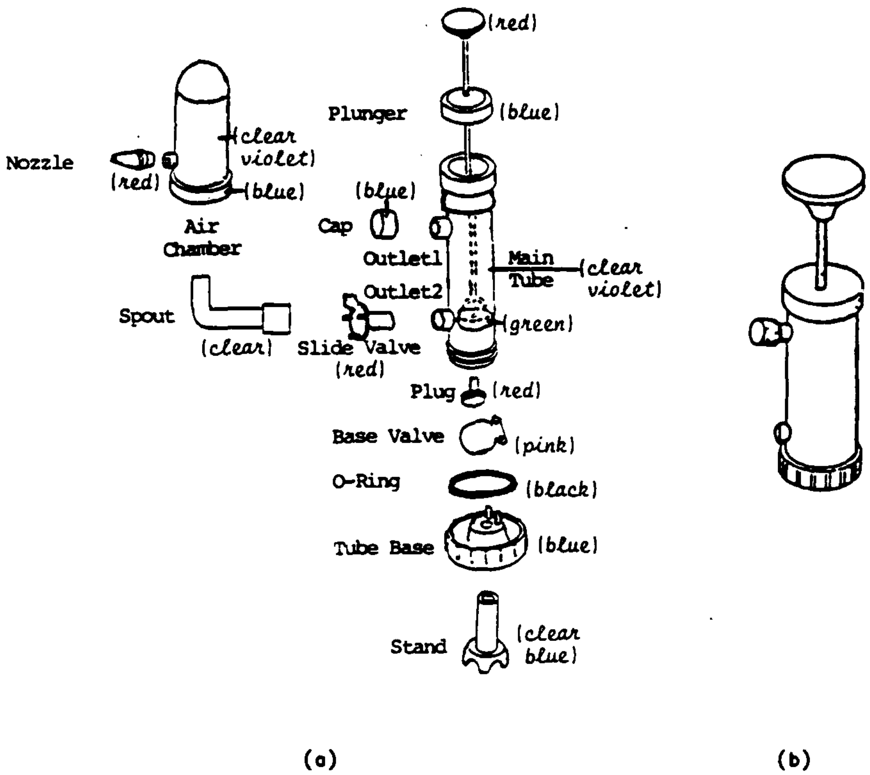


Figure 1-3: The Toy Water Pump

The example illustrates the complexity of reference indentification in a task-oriented domain. It shows that people do not always give up when a speaker's description isn't perfect but that they try to plow ahead anyway. The rest of this report will formalize the kinds of problems that occur during reference and then extend the reference paradigm to get around many of the problems.

1.4 The approach to the problem

I approach the issues mentioned in the previous sections from the perspective of a listener trying to interpret what he has just heard from a speaker. In this thesis, I present computer programs and algorithms that will play the part of the hearer. Since speakers are typically casual in how they form utterances, any computer hoping to play that part must have the same abilities for robust understanding that people do. Thus, it must be capable of taking what the speaker says and either delete, adapt or clarify it. This thesis concentrates on one aspect of this problem – the identification of referents for extensional descriptions and recovery from failed reference.

This thesis makes several claims about communication and miscommunication, about detecting and recovering from miscommunication, and especially about miscommunication due to reference failure.

1. Communication involves a great deal of miscommunication. Utterances often exhibit vagueness or errors. I develop in this work a taxonomy of situations where listeners typically get confused. If a natural language system is designed to expect such errors, then it, like people, can frequently recover from them. I show that enough structure often exists in the linguistic and physical context to indicate that a speaker has miscommunicated and to allow recovery from the miscommunication.

2. Reference identification is more than finding a referent or failing. I demonstrate that reference identification isn't the simple task assumed in past research, and I correctly find referents for descriptions that previous reference systems could not handle. In particular, I interpret noun phrases in a spatial world using real language. I show that such descriptions of objects can be vacuous because they are dependent on discourse context.

3. Knowledge about language and the world interacts with knowledge about reference. Listeners use their ability to distinguish feature values and their knowledge about the world to assign importance to parts of a speaker's description. They use these metrics to order features and then to selectively search the world for a referent.

4. Partial matching brings a listener one step closer to finding a referent for a failed description. Blind inexact matching of a failed description to objects in the world isn't sufficient to find the referent. In fact, often the closest match isn't even the best one. Partial matching does, however, provide a set of reasonable referent candidates. A more orderly way to determine the referent is proposed using a variety of knowledge sources like linguistic, perceptual, discourse, and trial and error.

5. Rule-based relaxation of the speaker's description provides a methodical way of finding a plausible referent. Rules were written to reflect many of my observations from analyzing the water pump protocols. These rules correspond to a subset of the the knowledge sources people draw on when performing reference. A control structure is required that determines how to apply the rules since the order in which rules are applied affects the outcome.

The rest of this thesis substantiates the above claims by describing a set of programs and algorithms that were developed to simulate a theory of reference identification for extensional descriptions and recovery from failed reference.

1.5 Overview of thesis organization

This thesis is divided into seven chapters.

Chapter 2 highlights some aspects of normal communication and then provides a general discussion on the types of miscommunication that occur in conversation, concentrating primarily on reference problems and motivating many of them with illustrative protocols.

Chapter 3 describes the process of reference identification, discussing the work by others in the area. Three natural language understanding systems are described.

Chapter 4 motivates a new paradigm for reference identification. A description of the program that I wrote to perform reference identification is also found here.

Chapter 5 illustrates the kinds of knowledge that people use in performing the reference task and describes rules for recovering from some failures of reference.

Chapter 6 presents some methods of attacking miscommunication in reference. Motivated here is a partial implementation of the relaxation component, FWIM, illustrated in Figure 1-1(b). It interprets many problematic referential descriptions.

Chapter 7 summarizes the goals and accomplishments of the work as well as providing some suggestions for future research.

The appendices contain introductory material and sample program runs. Appendix A shows how actions in the water pump domain are represented in KL-One. Appendix B shows a sample run of the parser and semantic interpreter. Appendix C provides a description and demonstration of the focus mechanism. Appendix D shows how comparatives, superlatives, and complex relations are handled in the system. Appendix E shows the basic reference system in action. Finally, Appendix F shows how the system explores for referent candidates.

2. MISCOMMUNICATION

This chapter provides a general discussion on miscommunication, describing the types that occur in conversation, and relating miscommunication to aspects of normal communication. It concentrates especially on miscommunication due to reference failures and motivates the discussion with illustrative protocols.[1]

2.1 Introduction

People must and do manage to resolve lots of (potential) miscommunication in everyday conversation. Much of it seems to be resolved subconsciously – with the listener unconcerned that anything is wrong. Other miscommunication is resolved with the listener actively deleting or replacing information in the speaker's utterance until it fits the current context. Sometimes this resolution is postponed until the questionable part of the utterance is actually needed. Still, when all these fail, the listener can ask the speaker to clarify what was said.[2]

The speaker often counts on the listener's ability to resolve minor problems and tends to be casual in his communication (e.g., see [14] on imprecise language.). The speaker, however, will become more careful in how his utterances are constructed when the cost of making mistakes becomes prohibitive. (For example, if mission control were sending instructions to astronauts in an orbiting space shuttle on how to replace tiles on the heat shield, they wouldn't mince words.) The costs vary with respect to the complexity of the task being communicated, the modality of communication (the bandwidth of the mode of communication affects how complete the speaker must be [52, 65] – e.g., in face-to-face mode, where the speaker can see the results of the listener's actions and correct or fine-tune them as appropriate, it is less costly to be careless) and the amount of task-specific expertise the speaker believes the listener possesses.

[1] Most of the examples in this section are taken from the water pump domain. Some, however, are from a set of dialogues in a graphics domain concerned with editing KL-One concepts on a display [70].

[2] An analysis of clarification subdialogues can be found in [41, 42].

There are many aspects of an utterance that the listener can become confused about and that can lead to miscommunication. The listener can become confused about what the speaker intends for the objects, the actions, and the goals described by the utterance. Confusions often appear to result from conflict between the current state of the conversation, the overall goal of the speaker, and the manner in which the speaker presented the information. However, when the listener steps back and is able to discover what kind of confusion is occurring, then the confusion can quite possibly be resolved.

2.2 Causes of miscommunication

Miscommunication is a part of normal communication because it happens so often. It is the result of confusions, failed expectations or assumptions on the part of the listener. It can be the consequence of the speaker being too haphazard in his construction of utterances, lapsed attention on the part of the listener, or inadequate shared knowledge. It is affected by environmental factors (such as the modality of communication used), the purpose of the communication, expectations of the listener on how the speaker will conduct himself when things go awry, and the knowledge the listener brings to the task. As I said earlier, in this work, there is a concentration on task-oriented dialogues. These conversations are between two people, one who is trying to explain to the other how to perform the task. Sometimes the participants are equally skilled (or unskilled) and they pool their resources to get something accomplished. This section motivates a paradigm for the kinds of conversation that I studied and points out places in the paradigm that leave room for miscommunication.

2.2.1 Environments that breed miscommunication

Miscommunication occurs more frequently in certain environments. Ochsman and Chapanis [52] found in their studies that the communication channel provided between two participants affects the ability of the participants to jointly solve a problem. The richer the communication mode, the easier and quicker it is for the participants to solve a problem (though face-to-face vs. voice were almost the same); the more limited it is, the more likely it is that mistakes will occur that cannot be easily detected and corrected. In cooperative problem solving tasks, where one of the dialogue participants is advising the other participant, the speaker doing the advising must take into account the communication channel when deciding how completely to formulate his utterances. If the speaker can see the results of the listener's actions, he can use less explicit instructions – filling in the details when the listener errs [52, 65]. If there is no visual feedback, however, the speaker must increase the details in each instruction or add utterances that request confirmation from the listener that an instruction was properly performed. When a speaker doesn't provide rich descriptions and checks in a limited communication channel, miscommunication is likely to result.

The task being discussed in the conversation also influences the likelihood of miscommunication occurring. A task composed of complex actions and goals increases the load put on the speaker to communicate successfully to the listener the goal he has in mind and the actions to be performed. Some features of the task that increase the likelihood of trouble are the number of steps in an action or plan, the need for the use of tools to accomplish a task, and the similarity between actions or plans in the set of known actions and plans available for accomplishing the task.

The objects in the environment have to be distinguished from each other to prevent confusion. Physical features of an object provide the primary way for people to discriminate objects [53]. If the objects in an environment have similar appearances, then it becomes more difficult to distinguish them. Functional information (information about how an object is used) can often provide a way to make a particular object and other contenders dissimilar [32].

2.2.2 Effects of the structure of task–oriented dialogues

Task–oriented conversations have a specific goal to be achieved: the performance of a task (e.g., the air compressor assembly in [30]). The participants in the dialogue can have the same skill level and they can work together to accomplish the task; or one of them, the expert, could know more and could direct the other, the apprentice, to perform the task. I have concentrated primarily on the latter case – due to the protocols that I examined – but many of my observations can be generalized to the former case, too.

The viewpoints of the expert and apprentice differ greatly in apprentice-expert exchanges. The expert, having an understanding of the functionality of the elements in the task, has more of a feel for how the elements work together, how they go together, and how the individual elements can be used. The apprentice normally has no such knowledge and must base his decisions on perceptual features such as shape [32].

The structure of the task affects the structure of the dialogue [30], particularly through the center of attention of the expert and apprentice during the accomplishment of each step of the task. The common center of attention of the dialogue participants is called the focus [30, 57, 69]. Shifts in focus correspond to shifts between th.: tasks and subtasks; e.g., the objects in a task and the subpieces of each object. Focus is represented by Grosz [30] as a piece of semantic network shared by both the speaker and and listener. Focus and focus shifts are governed by many rules [30, 57, 69]. For example, a focus shift can be directly stated (e.g., "what do I do next?") or it can be indirectly hinted at by pieces of an utterance (e.g., "the other one"). Confusion may result when expected shifts do not take place. For example, *if the expert changes focus* to some object but never bothers to talk about the object reasonably soon after its introduction (i.e., between the time of its introduction and its use, without digressing in a well-structured way in between (see [57])), or never discusses its subpieces (such as an obvious attachment surface), then the apprentice may become confused, leaving him ripe for miscommunication. The reverse influence between focus and objects can lead to trouble, too. A shift in focus by the expert that does not have a manifestation in the apprentice's world will also perplex the apprentice.

Focus also influences how descriptions are formed [32, 5]. The level of detail required in a description depends directly on the elements currently in focus. If the object to be described is similar to other elements in focus, the expert must be more specific in the formulation of the description or may consider shifting focus away from the confusing objects.

2.2.3 Discrepancies in knowledge and miscommunication

Just as with discrepancies in focus, discrepancies in knowledge between the speaker and listener can cause miscommunication. These disagreements can occur because the listener does not bring sufficient knowledge to the task and the speaker fails to convey enough information in his utterances to bring the listener up to a level of knowledge sufficient to perform the task. The speaker and listener could also have different beliefs. For example, differences between speaker and listener, such as what each believes about the other, can lead to false assumptions that each may use when interpreting the other's utterances. Knowledge differences, though, can sometimes provide a means to help detect miscommunication. For example, a listener's knowledge about the world in which the task is taking place can provide a way of checking whether or not a speaker's utterance is realistic. The listener can simply examine the world and compare it to the speaker's utterance or try to do what the speaker requests and see if it succeeds.

2.2.3.1 Knowledge the listener brings to the task

In apprentice-expert dialogues such as those in the water pump domain, the knowledge brought to the task by a naive apprentice is limited to four principal areas: (1) language abilities, (2) perceptual abilities to identify objects, (3) past experience and knowledge in assembling objects, and (4) the ability to perform trial-and-error tests in the real world. The language abilities of the apprentice allow him to follow the flow of information provided by the expert in his utterances and descriptions. Syntactic, semantic and pragmatic knowledge compose this knowledge about language. A more detailed description of these knowledge sources can be found in [63] and in Chapter 5.

Perceptual abilities include the recognition of physical features of an object such as size, shape, color, location, composition and transparency. The fineness of each category's partitioning varies among individuals. For example, some people know more color values than others. An expert, if he wishes to prevent misreference, may choose to use only basic level descriptions in each category until the apprentice demonstrates a broader knowledge or the expert can familiarize the apprentice with other values.

The past experience someone has with objects provides a method for the expert to tie a description down to a common point of view. If an object has a familiar name, the expert can refer to it by that name. The expert can also refer by making analogies to everyday objects as a model for the apprentice in his selection of a referent. The analogies can be through the shapes or functions of everyday objects. The same holds true for actions – past experience makes it easier for the expert to describe an action to the apprentice.

Finally, the apprentice brings to a task the ability to perform simple tests. He can experiment to determine whether two pieces can be attached. In the water pump domain, attachment is performed by pushing, twisting or screwing one object into or onto another. During and after the attachment process, one can determine how good a fit is by noting the compatibility of the shapes of the attaching surfaces (and this can be used to align the surfaces) and by checking the snugness of the fit once the objects are attached.

2.2.3.2 The knowledge transferred in an utterance

In an apprentice–expert domain there is limited shared knowledge between speaker and listener, less so than in many other domains since usually one participant knows a lot more about the task than the other. This requires a transfer of knowledge from the speaker – who is explaining how to perform the task – to the listener – who is to perform the task. The listener, thus, is building up knowledge (which becomes shared or mutually believed knowledge [53, 51, 17, 55, 38, 50]) from the speaker's utterances while attempting to perform the task.

At least two kinds of knowledge are conveyed in an utterance. For this paper I will focus on task knowledge and communicative knowledge. Task knowledge is knowledge about the specific domain that is used to fill the propositional content of an utterance. It refers to three kinds of things in the water pump domain: (1) the objects, the set of parts available to accomplish the task (i.e., the "real world" which is the physical environment around the conversational participants); (2) the actions, the set of physical actions available to the listener; and (3) instructions linking objects and actions together to achieve some goal. Communicative knowledge consists of speech acts, communicative goals, and communicative actions. Speech acts are underlying forms that are performed by the speaker in expressing an utterance (e.g., REQUEST, INFORM) [67, 18, 3]. They provide an illocutionary force that is applied to the proposition expressed in an utterance. Communicative goals reflect the structure of the discourse (e.g., setting up a topic, clarifying, or adding more information [4]). They express how an utterance is to be understood with respect to the high-level communicative goals reflected in the structure of the dialogue and hence how the task the utterance examines is performed. A communicative act is a way of accomplishing the communicative goal that one wants to convey (e.g., communicate the goal, communicate the object's description, communicate the action). Only some of the possible communicative acts may be reasonable at any one time to accomplish the current communicative goal [58, 4, 40].

Miscommunication can occur due to the way the information was transferred (e.g., communicative knowledge) or the content of what was transferred (e.g., task knowledge). Task knowledge-based miscommunication occurs when the speaker is unaware that (1) the listener has a different view of the task, (2) the listener is considering a different subset of objects, (3) the listener is considering a different subset of actions, and so on. Difficulties with communicative knowledge are also possible. The speaker may use the wrong speech act (e.g., utters something (inadvertently) that would be conventionally interpreted as an INFORM when meant as a REQUEST) or the listener errs when interpreting the speaker's intention (e.g., the speaker may be INFORMing the listener that the blue cap fits around the end of the tube but the listener might interpret the utterance as a REQUEST to actually place the cap around the end of the tube). In both cases it is the effect of the speech act that causes the trouble since it influences what the listener will do with what was said (i.e., determine the intended responses). Finally, communicative knowledge can cause mistakes and confusion if the listener and speaker differ on the communicative goal (e.g., the listener might think the speaker is clarifying previous information when, in fact, the speaker is adding new information). They will feel they are communicating at cross purposes – leading to frustration.

2.3 Instances of miscommunication

In this section I will present evidence that people do miscommunicate and yet they manage to repair reference failures. I will look at specific forms of miscommunication and describe ways to detect them. I will highlight relationships between different miscommunication problems and will demonstrate ways for resolving some of them. A common thread in many of the miscommunications revolves around the degree of specificity of the speaker's utterances.

There are many ways hearers can get confused during a conversation. Figure 2-1 outlines some of them that were derived from analyzing the water pump protocols. This section defines and illustrates many of the confusions in the taxonomy through numerous excerpts. Each excerpt has marked in parentheses the modality of communication that was used in the excerpt (face-to-face, over the telephone, and so forth). A description about the collection of these excerpts can be found in [20]. Each bracketed portion of the excerpt explains what was occurring at that point in the dialogue. The confusions themselves, coupled with the description at the end of this chapter on how to recognize when one of them is occurring, provide motivation for the use of the algorithm outlined in Chapter 5 and 6 as a means for repairing communication problems. Another categorization of confusions that lead to conversation failure can be found in [60].

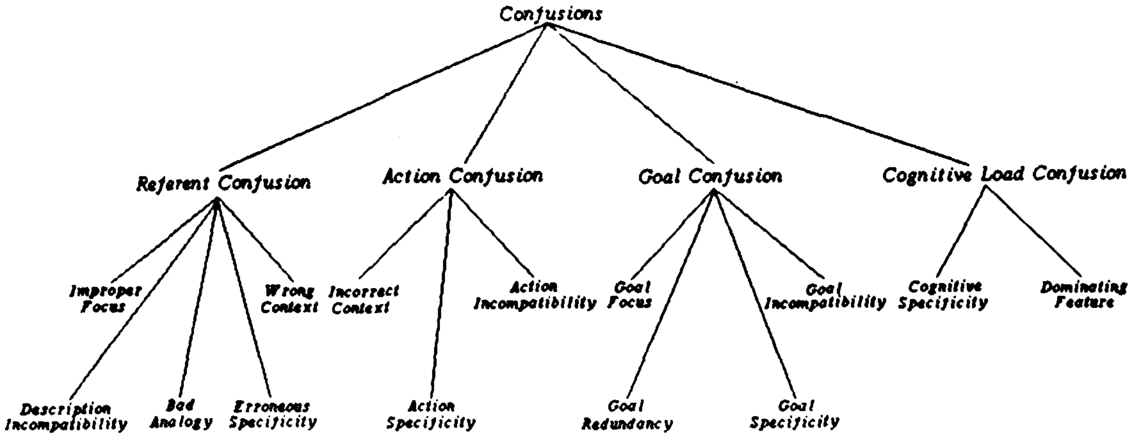


Figure 2-1: A taxonomy of confusions

2.3.1 Referent confusion

Referent confusion occurs when the listener is unable to determine correctly what the speaker is referring to with a particular description.[3] It may occur when the descriptions in the utterance are ambiguous or imprecise, when there is confusion between the speaker and the listener about what the current focus or context is, or when the descriptions in the utterance are either incorrect or incompatible with the current or global context.

Erroneous Specificity

A speaker can be underspecific or overspecific in his descriptions. Such descriptions are a form of erroneous specificity that can lead to mistakes on the part of the listener even though, technically, nothing is wrong with the description.

Ambiguous descriptions are underspecified and can cause confusion about the referent. Excerpt 2 below illustrates a case where the speaker's description is underspecified - it does not provide enough detail to prune the set of possible referents down to one.

Excerpt 2 (Face-to-Face)

S: 1. And now take the little red
2. peg,
[P takes PLUG]
3. Yes,

4. and place it in the hole at the
5. green end,
[P starts to put PLUG into OUTLET2 of MAINTUBE]
6. no

7. the--in the green thing
[P puts PLUG into green part of PLUNGER]

P: 8. Okay.

[3] See [81, 15, 59, 51, 30, 78, 69, 58, 27, 28]for introductory discussions on the identification of referents.

In Line 4 and 5, S describes the location to place a peg into a hole by giving spatial information. Since the location is given relative to another location by "in the hole at the green end", it defines a region where the peg might go instead of a specific location. In this particular case, there are three possible holes to choose from that are near the green end. The listener chooses one – the wrong one – and inserts the peg into it. Because this dialogue took place face to face, S is able to correct the ambiguity in Lines 6 and 7.

An underspecified description can be imprecise in many possible ways. I will mention a couple of them. (1) A description may consist of features that do not readily apply or that are inappropriate in the domain. In Line 3, Excerpt 3, the feature "funny" has no meaning to the listener here. It is not until A provides a fuller description in Lines 5 to 8 that E is able to select the proper piece. (2) It may use imprecise feature values. For example, one could use an imprecise head noun coupled with few or no feature values (and context alone does not necessarily suffice to distinguish the object). In Excerpt 4, Line 9, "attachment" is imprecise because all objects in the domain are attachable parts. The expert's use of "attachment" was most likely to signal the action the apprentice can expect to take next. The use of the feature value "clear" provides little benefit either because three clear, unused parts exist. The size descriptor "little" prunes this set of possible referents down to two contenders. Another use of imprecise feature values occurs when enough feature values are provided but at least one value is too imprecise. In Excerpt 5, Line 3, the use of the attribute value "rounded" to describe the shape does not sufficiently reduce the set of four possible referents (though, in this particular instance, A correctly identifies it) because the term is applicable to numerous parts in the domain.[4] A more precise shape descriptor such as "bell-shaped" or "cylindrical" would have been more beneficial to the listener.

Excerpt 3 (Telephone)

E: 1. All right.

[4] "Chamber" was interpreted here in a broader sense by the listener because it was used right at the beginning of the dialogue. This was before the speaker introduced other terms such as "tube" that would have helped distinguish the pieces better. The example demonstrates how discourse affects reference.

2. Now.

3. There's another funny little
4. red thing, a

[A is confused, examines both NOZZLE and SLIDEVALVE]

5. little teeny red thing that's
6. some--should be somewhere on
7. the desk, that has um--there's
8. like teeth on one end.

[E takes SLIDEVALVE]

A: 9. Okay.

E: 10. It's a funny-loo--hollow,
11. hollow projection on one end
12. and then teeth on the other.

Excerpt 4 (Teletype)

A: 1. take the red thing with the
2. prongs on it

3. and fit it onto the other hole
4. of the cylinder

5. so that the prongs are
6. sticking out

R: 7. ok

A: 8. now take the clear little
9. attachment

10. and put on the hole where you
11. just put the red cap on

12. make sure it points
13. upward

R: 14. ok

Excerpt 5 (Teletype)

S: 1. Ok,

2. put the red nozzle on the outlet
3. of the rounded clear chamber

4. ok?

A: 5. got it.

A description is overspecific if it contains a feature value that is so specific that it is hard for the listener to verify that a particular referent candidate exhibits that value. For example, a listener may be told to pick up the "chartreuse tube" but isn't really sure that the "green tube" he sees is it because he doesn't know enough different shades of green. He might even know that chartreuse is a kind of yellowish-green but that isn't good enough for him to recognize it. Other examples of overspecificity can be found in the sections on Bad Analogy and Cognitive Specificity.

Improper Focus

Earlier I talked about focus and problems that occur due to it. In this section, I discuss how misfocus can cause misreference. Focus confusion can occur when the speaker sets up one focus and then proceeds with another one without letting the listener know of the switch (i.e., a focus shift occurs without any indication). The opposite phenomenon can also happen – the listener may feel that a focus shift has taken place when the speaker actually never intended one. These really are very similar – one is viewed more strongly from the perspective of the speaker and the other from the listener.

Excerpt 6 below illustrates an instance of the first type of focus confusion. In the excerpt, the speaker (S) shifts focus without notifying the listener (P) of the switch. As the excerpt begins, P is holding the *TUBEBASE*. S provides in Lines 1 to 16 instructions for P to attach the *CAP* and the *SPOUT* to outlets *OUTLET1* and *OUTLET2*, respectively, on the *MAINTUBE*. Upon P's successful completion of these attachments, S switches focus in Lines 17 to 20 to the *TUBEBASE* assembly and requests P to screw it on to the bottom of the *MAINTUBE*. While P completes the task, S realizes she left out a step in the assembly – the placement of the *SLIDEVALVE* into *OUTLET2* of the *MAINTUBE* before the *SPOUT* is placed over the same outlet. S attempts to correct her mistake by requesting P to remove "the plas"[5] piece in Lines 22 and 23. Since S never indicated a shift in focus from the *TUBEBASE* back to the *SPOUT*, P interprets "the plas" to refer to the *TUBEBASE*.

Excerpt 6 (Face-to-Face)

S: 1. And place
2. the blue cap that's left
[P takes CAP]
3. on the side holes that are
4. on the cylinder,
[P lays down TUBEBASE]
5. the side hole that is farthest
6. from the green end.
[P puts CAP on OUTLET1 of MAINTUBE]

P: 7. Okay.

S: 8. And take the nozzle-looking
9. piece,
[P grabs NOZZLE]

10. no

11. I mean the clear plastic one,
[P takes SPOUT]

12. and place it on the other hole
[P identifies OUTLET2 of MAINTUBE]
13. that's left,

14. so that nozzle points away
15. from the
[P installs SPOUT on OUTLET2 of MAINTUBE]

16. right.

P: 17. Okay.

[5] The whole word here is "plastic." In these protocols, people often guess before hearing the whole utterance or even whole words.

S: 18. Now

19. take the

20. cap base thing

[P takes TUBEBASE]

21. and screw it onto the bottom,

[P screws TUBEBASE on MAINTUBE]

22. ooops,

[S realizes she has forgotten to have P put SLIDEVALVE into OUTLET2 of MAINTUBE]

23. un–undo the plas

[P starts to take TUBEBASE off MAINTUBE]

24. no

25. the clear plastic thing that I
26. told you to put on

[P removes SPOUT]

27. sorry.

28. And place the little red thing

[P takes SLIDEVALVE]

29. in there first,

[P inserts SLIDEVALVE into OUTLET2 of MAINTUBE]

30. it fits loosely in there.

Excerpt 7 below demonstrates the latter type of focus confusion that occurs when the speaker (S) sets up one focus – the *MAINTUBE*, which is the correct focus in this case – but then proceeds in such a manner that the listener (J) thinks a focus shift to another piece, the *TUBEBASE*, has occurred. Thus, Line 15 refers to "the lower side hole in the *MAINTUBE*" for S and "the hole in the *TUBEBASE*" for J. J has no way of realizing that he has focussed incorrectly unless the description as he interprets it doesn't have a real world correlate (here something does satisfy the description so J doesn't sense any problem) or if, later in the exchange, a conflict arises due to the mistake (e.g., a requested action can not be performed). In Line 31, J inserts a piece into the wrong hole because of the misunderstanding in Line 15. Line 31 hints that J may have become suspicious that an ambiguity existed somewhere in the previous conversation but since the task appeared to be successfully completed (i.e., the red piece fit into the hole in the base), and since S did not provide any clarification, he assumed he was correct.

Excerpt 7 (Telephone)

S: 1. Um now.
2. Now we're getting a little
3. more difficult.

J: 4. (laughs)

S: 5. Pick out the large air tube
[J picks up STAND]
6. that has the plunger in it.
[J puts down STAND, takes PLUNGER/MAINTUBE assembly]

J: 7. Okay.

S: 8. And set it on its base,
[J puts down MAINTUBE, standing vertically, on the *TABLE*]
9. which is blue now,
10. right?
[J has shifted focus to the TUBEBASE]

J: 11. Yeah.

S: 12. Base is blue.
13. Okay,
14. Now
15. You've got a bottom hole still
16. to be filled,
17. correct?

J: 18. Yeah.
[J answers this with MAINTUBE still sitting on the TABLE; he shows no indication of what hole he thinks is meant – the one on the MAINTUBE, OUTLET2, or the one in the TUBEBASE]

S: 19. Okay.
20. You have one red piece

21. remaining?

[J picks up MAINTUBE assembly and looks at TUBEBASE, rotating the MAINTUBE so that TUBEBASE is pointed up, and sees the hole in it; he then looks at the SLIDEVALVE]

J: 22. Yeah.

S: 23. Okay.
24. Take that red piece.

[J takes SLIDEVALVE]

25. It's got four little feet on
26. it?

J: 27. Yeah.

S: 28. And put the small end into
29. that hole on the air tube--

30. on the big tube.

J: 31. On the very bottom?

[J starts to put it into the bottom hole of TUBEBASE – though he indicates he is unsure of himself]

S: 32. On the bottom,
33. Yes.

Misfocus can also occur when the speaker inadvertently fails to distinguish the proper focus because he did not notice a possible ambiguity; or when, through no fault of the speaker, the listener just fails to recognize a switch in focus indicated by the speaker. Excerpt 7 above is an example of the first type because S failed to notice that an ambiguity existed since he never explicitly brought the *TUBEBASE* either into or out of focus. He just assumed that J had the same perspective as he had – a perspective in which no ambiguity occurred.

Wrong Context

Context differs from focus. The context of a portion of a conversation is concerned with the intention of the discussion in that fragment and with the set of objects relevant to that discussion, though not attended to currently. Focus pertains to the elements which are currently being attended to in the context. For example, two people can share the same context but have different focus assignments within it – we're both talking about the water pump but you're describing the *MAINTUBE* and I'm describing the *AIRCHAMBER*. Alternatively, we could just be using different contexts – I think you're talking about taking the pump apart but you're talking about replacing the pump with new parts – in both cases we may be sharing the same focus – the pump – but our contexts are totally off from one another.[6] The kinds of misunderstandings that can occur because of context inconsistencies are similar to those for focus problems: (1) the speaker might set up or use one context for a discussion and then proceed in another one without effectively letting the listener know of the change, (2) the listener may feel a change in context has taken place when in fact the speaker never intended one, or (3) the listener fails to recognize an indicated context switch by the speaker. Context affects reference identification because it helps define the set of available objects that are possible contenders for the referent of the speaker's descriptions. If the contexts of the speaker and listener differ, then misreference might result.

Bad Analogy

An analogy (see [24] for a discussion on analogies) is a useful way to help describe an object by attempting to be more precise by using shared past experience and knowledge – especially shape and functional information. If that past experience or knowledge doesn't contain the information the speaker assumes it does, then trouble occurs. Thus, one more way referent confusion can occur is by describing an object using a poor analogy.

An analogy can be improper for several reasons. It might not be specific enough – confusing the listener because several potential referents might conform to the analogy. Alternatively, the analogy might fail because discovering a mapping between the analogous object and something in the environment is too difficult. In Excerpt 8, J at first has trouble correctly satisfying A's functional analogy "stopper" in "the big

[6] Grosz [30, 32] would describe this as a difference in "task plans" while Reichman [57, 58] would say that the "communicative goals" differed.

blue stopper", but finally selects what he considers to be the closest match to "stopper". The problem for J was that A's functional analogy was not specific enough. It would have been better to use "cap" instead of "stopper."

Excerpt 8 (Telephone)

A: 1. Okay. Now,

2. take the big blue
3. stopper that's laying around

[J grabs AIRCHAMBER]

4. ... and take the black
5. ring--

J: 6. The big blue stopper?

[J is confused and tries to communicate it to A; he is holding the AIRCHAMBER here]

A: 7. Yeah,

8. the big blue stopper

9. and the black ring.

[J drops AIRCHAMBER and takes the O-RING and the TUBEBASE]

In other cases the analogy might be too specific - confusing the listener because none of the available referents appear to fit it. In Line 8 of Excerpt 6, "nozzle-looking" forms a poor shape analogy because the object being referred to actually is an elbow-shaped spout and not a nozzle. The "nozzle-looking" part of the description convinced the listener that what he was looking for was something identified by the typical properties of a nozzle (which is a small tube used as an outlet). However, sometimes when an object is a clear representative of a specified analogy class, the apprentice will not tend to select it as the intended referent. He would assume that, to refer to that object, the expert would not bother to form an analogy instead of just directly describing the object as a member of the class. Hence, the apprentice may very well ignore the best representative of the class for some less obvious exemplar. Given the case just mentioned, it is therefore better to say "nozzle" instead of "nozzle-looking." In Excerpt 9, the description "hippopotamus face shape" (a shape analogy) in Lines 2 and 3, and "champagne top" (a shape analogy) in Line 9, are too specific and the listener is unable to easily find something close enough to match either of them. He can't discover a mapping between the object in the analogy and one in the real world (a discussion on discovering such mappings can be found in [24]). In fact, when this excerpt was played back to one listener, he was so overwhelmed by M's descriptions, that he exclaimed "What!" when he heard them and was unable to correctly proceed.

Excerpt 9 (Audiotape)

M: 1. take the bright pink flat
2. piece of hippopotamus face
3. shape piece of plastic
4. and you notice that the two
5. holes on it

[M is trying to refer to BASEVALVE]

6. match
7. along with the two
8. peg holes on the
9. champagne top sort of
10. looking bottom that had
11. threads on it

[M is trying to refer to TUBEBASE]

Description Incompatibility

Descriptions incompatible with the scene can lead to confusion also. A description is incompatible when it does not agree with the current state of the world: (1) when one or more of the specified conditions, i.e., the feature values, do not satisfy <u>any</u> of the pieces; (2) when one or more specified constraints do not hold (e.g., saying "the <u>loose</u> one" when all objects are tightly attached); or (3) if no <u>one</u> object satisfies <u>all</u> of the features specified in the description. In Lines 7 and 8 of Excerpt 9 above, M's description of "the two peg holes" leads to bewilderment for the listener because the "champagne top sort of looking bottom that had threads on it" (i.e., the *TUBEBASE*) has no holes in it. M actually meant "two pegs". The use of "peg" and "hole" interchangeably and other similar word pairs (see [30] for more of them) are often tolerated by listeners who recognize such object pair confusions.

2.3.2 Action confusion

Actions in the water pump domain are simple enough that few confusions occur. However, for more complex actions, one could expect that the apprentice would be unsure what tools to use, how to use a tool, what order to carry out the steps in an action, and when a task is (successfully) completed. Such an environment is ripe for confusions similar to those that occur for referents: specificity, context, and incompatible action. These are described in detail below.

Action Specificity

Action specificity confusion can result in the listener being unable to perform a requested action or even performing the wrong action. It can occur when the speaker's description is underspecified, not providing enough detail to prune the set of possible actions down to one. It can also occur when the description of the action is so imprecise that it is impossible to determine what the speaker wants done or how to do it. In Lines 10 and 11 of Excerpt 10 below, J requests a more precise description of the action, "put it over that bottom opening, too", requested in Lines 5 and 6. Here J was confused about how to perform the action with the specified object.

Excerpt 10 (Telephone)

A: 1. Okay,

2. then take that clear
3. plastic elbow joint.

[J takes SPOUT]

J: 4. All right.

A: 5. And put it over that bottom
6. opening, too.

J: 7. (pause) Okay.

A: 8. Okay.

9. Now, take the--

J: 10. Which end am I supposed to put
11. it over? Do you know?

A: 12. Put the--put the--the big end--

the big end over it.

J: 13. All right.

Incorrect Context

Context confusion for actions, is just like context confusion for referents, and can occur when the speaker sets up one context and then proceeds in another one without notifying the listener, when the speaker fails to distinguish the proper context because he didn't notice a possible ambiguity, or when the listener fails to notice (or ignores) a switch in context or misanticipates what the new context will be. For example, the speaker could request the listener to carry out a particular action (e.g., an attachment) to an object using a specific tool. Upon completion of that action, the speaker could request the listener to attach another object. If the new object requires a different tool to attach it and the speaker hasn't made that clear, one would expect the listener to initially try to attach the object with the current tool. Only after an obstacle occurs will the listener question the use of the tool.

Action Incompatibility

Confusion can occur due to the incompatibility of an action with respect to past requests by the speaker. This requires comparing the current action to just completed actions and considering the result of performing the current action. A listener must investigate whether or not the current action was successfully completed (which can itself be hard to judge). The listener can also determine if any specified constraints failed. In Lines 16 to 18 of Excerpt 11 below, B complains that a constraint associated with K's requested action in Lines 11 to 15, that the bell jar fit over the red valve (i.e., the *SLIDEVALVE*), fails.[7] This causes confusion that B resolves in Lines 16 to 18 when he discovers a piece, the elbow (i.e., the *SPOUT*), that will fit over the *SLIDEVALVE*. Notice that it is the nonperformance of the action here that sparks B's confusion and not the inability of B to find the object's referent from its description (i.e., B finds the proper referent as described by K in Lines 1 and 2 - the

[7] Actually, K told B the wrong thing to do here.

AIRCHAMBER). Thus, this is an example of an incompatible action problem instead of an object referent problem. Often, getting around such problems requires the listener to stop and ask for clarification from the speaker. Sometimes, however, the listener will use trial and error techniques to see if he can find another object that works with the requested action or will try some related action on the requested object. Such techniques are compatible with a plan-based account of language and action.

Excerpt 11 (Teletype)

K: 1. Then take the piece with the blue
2. base and the clear glass cover

3. The hole in the blue base goes on
4. the side of the plastic tube that
5. we put the red thing in earlier.

6. Got it

B: 7. wait

8. You mean the small bell jar fits
9. directly on to the clear tube near
10. the red thing with spikes??

K: 11. right

12. the red thing with the spikes
13. should be in the hole on the side
14. of the big plastic tube.

15. the bell jar fits right over it.

B: 16. The only thing that I have that
17. will fit over the red thing is an
18. elbow pipe

K: 19. you're right

20. that should go on the side of the
21. plastic tube instead of the bell
22. jar

2.3.3 Goal Confusion

Goals are broader than actions, expressing what one would like to see achieved. They often have more to do with the intent behind the dialogue and less to do with the content of the current utterance. In some sense they form a framework to hang individual utterances on. They can reach beyond the tasks or actions capable of satisfying the goal and can have something to do with a speakers' or listeners' beliefs about their conversation partners. A speaker, thus, has to try to get the listener to come to the right set of beliefs about the speaker's goals. A speaker's goals and listener's goals can differ. The closer they are to each other, the easier it is to communicate, i.e., they will have less confusion and will not be working at cross purposes.

Goal specificity

Goal specificity has to do with how broad or narrow a goal is. A broad goal is usually imprecise. It points in a particular direction, but doesn't completely specify what is wanted. Speakers often underspecify their goals either because they are uncertain of them, or because they assume their conversational partner "knows what they mean." Their goal can often be satisfied by performing any number of actions. For example, requesting that <u>more</u> space be created on a graphics display is satisfiable by making the screen completely blank, moving the display upward, erasing segments of the display, buying a larger display, and so on. A narrow goal, however, is usually well-defined, clearly describing what is wanted. For example, following the above example, the goal of erasing item X from the display to make more room, is very specific when compared to the goal of making more room.

Goal specificity is measured with respect to the goals available in a particular domain (e.g., deciding if this goal is broad or narrow compared to other relevant goals) and the goals invoked earlier in a dialogue (e.g., noting shifts "from a narrow to a broad goal" or "from a broad to a narrow goal"). Goal specificity is a factor in whether or not a listener will become confused. For a particular goal, <u>if</u> the goal is too broad, there may not exist a unique plan that seems most applicable to satisfying the goal, or it may be hard to find one. When a goal is too narrow, then no plan may appear capable of achieving the goal because none of the plans seem related to the goal.

Goal focus

Goal focus is the actual intent behind a set of utterances. It is central to avoiding confusion. If the actual goal of the speaker differs from the perceived goal of the listener, tasks will not be successfully completed or will only be partially completed. In Excerpt 12 [70] below (also see Figure 2-2), U and S have not established a common goal focus in Lines 3 and 4. S assumed that U had one particular goal in mind - to draw on the display screen - while U really intended S to consider the deeper task of manipulating the underlying data base, too. The problem occurs because U was imprecise in setting up the goals to be achieved. Any natural language understanding system, hence, needs the ability to step back and assess what is going on when conflicts arise.

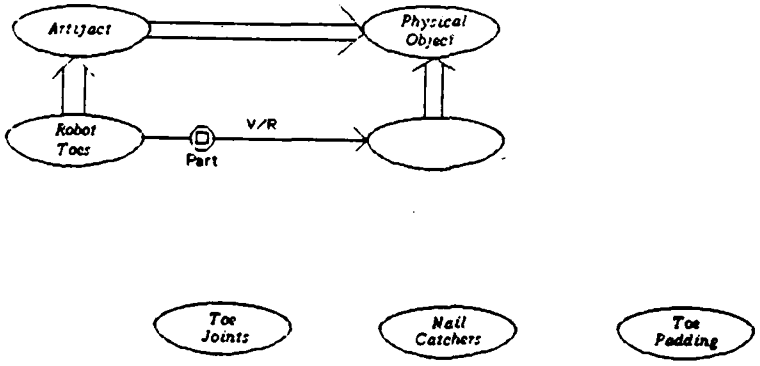


Figure 2-2: The Display

Excerpt 12 (Teletype with common graphics display)

U: 1. Good. Now put a part role on robot toes whose
2. VR is unlabelled and which is SUPERC'ed up
3. to physical objects, and under it put three
4. generics labelled toe joints, nail catchers, and
5. toe padding. That'll finish this little bit.

S. 6. Drawing (sigh)...Ok

U: 7. You forgot the cables

S: 8. You didn't ask for any

U: 9. Aaarrgh. What do you think I meant
10. by under?

S: 11. I thought you meant under the generic on
12. the screen. Was I wrong?

U: 13. Yes, I didn't realize you were so
14. bloody literal-minded.

Goal incompatibility

While conversations may have a single goal, usually such a goal would have to be very abstract. A conversation actually consists of a set or sequence of subgoals. These subgoals are incompatible if they are contradictory or discontinuous. This typically occurs when the speaker has misspoken or changed his mind, but can result from fundamental differences in the beliefs of the speaker and listener. Goal contradictions occur when the current subgoal does not mesh with the overall goal. Goal discontinuity occurs when the new subgoal does not fit past ones. Sometimes determining the incompatibility of two subgoals requires probing deeper into the intent behind them. In Excerpt 13 [70], where the context is that of adding information to the screen and data base, U requests in Line 1 that S delete information. S notes that requesting the deletion is contradictory with the current task of augmenting the data base, and requests in Line 4 and 5 clarification from U. Here S realizes deletion from the screen is reasonable since it reduces screen clutter but that deletion from the data base is not.

Excerpt 13 (Teletype with common graphics display)

U: 1. Sorry, while you're at it, you can delete the
2. concept for DZZ employees and for Israel
3. and RNAIL(?) and all their links, I think.

S: 4. Do you want these concepts deleted from
5. the data base? [CONFIRM]

U: 6. No, just from the picture

2.3.4 Cognitive load confusion

Cognitive properties can affect a listener's ability to comprehend. They may cause a listener to feel confused. A speaker who is overspecific or underspecific in his requests or who overemphasizes one part of the request can overload the listener cognitively – causing mistakes on the part of the listener even though, technically, nothing is wrong with the request. Other cognitive load problems that will not be considered here include the rate a speaker makes his requests, the use of complex or awkward grammatical constructions in a speaker's utterances, or the use of unfamiliar words in a speaker's utterances.

Cognitive Specificity

The manner in which a speaker presents his requests to the listener has a bearing on how well the listener comprehends them. A speaker can be overspecific [29, 32] or underspecific about the task he is asking to be performed. A request is overspecific if extra details are given that seem obvious to the listener [31]. Since the listener would not expect the speaker to provide him with obvious details, the listener might become confused that he had done something incorrectly as the task seemed easier than the one apparently described by the speaker.[8] For example, in Excerpt 14, S's description of the bubbled piece is overspecific because it supplies many more features than needed to identify the piece. The extra description in Lines 15 to 17 confused the listener who appeared to have correctly identified the piece by Line 13 but ended up taking the wrong one when the expert kept adding more details.

Excerpt 14 (Telephone)

S: 1. Okay?

2. Now you have two devices that
3. are clear plastic.

[J picks up MAINTUBE and SPOUT]

J: 4. Okay.

[8] Of course, there are some situations – such as teaching – where the hearer would be more willing to tolerate overspecific descriptions.

S: 5. One of them has two openings
6. on the outside with threads on
7. the end, and its about five
8. inches long.
[J rotates MAINTUBE confirming S's description]
9. Do you see that?

J: 10. Yeah.

S: 11. Okay,

12. the other one is a bubbled
13. piece with a blue base on it
14. with one spout.
[J looks at AIRCHAMBER]

15. Do you see it?

16. About two inches long.
[J picks up STAND and drops MAINTUBE]
17. Both of these are tubular.
[J puts down SPOUT]

J: 18. Okay.

19. not the bent one.
[J puts down SPOUT]

A request is underspecific if not enough details are given to make the listener feel he had correctly accomplished a task when the task was difficult to perform. For example, in Excerpt 15 below, C requests the listener to install the *PLUNGER* into the *MAINTUBE*. In describing how to install the *PLUNGER*, C mentions that the blue cap part of the *PLUNGER* fits very tightly, requiring a lot of force to put it on. Later in the conversation, C tells the listener to install the *BASEVALVE* over two prongs on the *TUBEBASE*. However, he neglects to warn the listener that the piece fits tightly into the holes. The listener, finding the fit to be very tight, becomes confused and suspects that he has made a mistake (either selecting the wrong piece or the wrong hole). The listener commented to the experimenter that he thought something was wrong because C had earlier been very careful to warn him when a fit was tight. A listener might try other pieces or holes in search of one that seems to work better.

Excerpt 15 (Written)

C: 1. Place the plunger into the top of the cylinder,
2. green end first, pushing it down until
3. the green cap is securely in. Fit
4. the blue cap onto the cylinder. It's
5. a tight fit so you have to force it.
6. . . .
7. On the table is a small pink tab
8. with two holes. Place the two
9. holes in the tab over the two prongs
10. on the cap.

[Listener/Reader becomes confused]

Dominating feature

Another way the speaker can confuse the listener is by using a dominating feature in a description. A dominating feature value can overpower a description, causing the listener to avoid attending to other feature values that are given. In Lines 2 and 3 of Excerpt 14 above, S describes a transparent tube that has a violet tint and a colorless, transparent tube by "two devices that are clear plastic". The feature value "clear" dominates the description making the listener assume the objects are also colorless.

2.4 Detecting miscommunication

Part of my research has been to examine how a listener discovers the need for a repair of an utterance or a description during communication. The incompatibility of a description or action with the scene is one signal of possible trouble. The appearance of a goal incompatibility such as an obstacle or redundancy that blocks one from achieving a goal is another indication of a potential problem.

2.4.1 Description and Action Incompatibility

As I pointed out in earlier sections, there are three kinds of possible incompatibility with the scene - description, action and goal. The strongest hint that there is a description incompatibility occurs when the listener finds no real world object to correspond to the speaker's description (i.e., referent identificatiun fails). This can occur when (1) one or more of the specified feature values in the description are not satisfied by any of the pieces (e.g., saying "the orange cap" when none of the objects are orange); (2) when one or more specified constraints do not hold (e.g., saying "the red plug that fits loosely" when all the red plugs attach tightly); or (3) if no one object satisfies all of the features specified in the description (i.e., there is, for each feature, an object that exhibits the specified feature value, but no one object exhibits all of the values). An impossible reference could indicate an earlier action error (e.g., two parts were put together that never had been intended to be assembled together). An action incompatibility problem is likely if (1) the listener cannot perfcrm the action specified by the speaker because of some obstacle; (2) the listener performs the action but does not arrive at its intended effect (i.e., a specified or default constraint isn't satisfied); or (3) the current action affects a previous action in an adverse way, yet the speaker has given no sign of any importance to this side-effect. Action incompatibility might indicate an earlier misference (e.g., you chose the wrong part and used it in an earlier action).

2.4.2 Goal obstacle

A goal obstacle occurs when a goal (or subgoal) one is trying to achieve is blocked. This blockage can result in confusion for the listener because in general listeners do not expect speakers to give them tasks that can not be achieved. Often, though, it points out for the listener that some miscommunication (such as misreference) has occurred. Excerpt 16 below shows an example where the goal of A cannot be achieved because an obstacle has occurred. Lines 5 to 8 indicate that the goal of A is for J to install the *STAND* into the *TUBEBASE* on the *MAINTUBE*. At Line 11, however, J indicates trouble has occurred in trying to achieve A's objective and he is requesting A to provide more detail. The actual trouble J encountered was that the spot in the *TUBEBASE* where the *STAND* was to go was already filled.

Excerpt 16 (Telephone)

A: 1. Okay, the big--okay. I know--
2. I think I know what you meant.
3. Um--you--you--that's supposed--
4. that uh clear plastic blue

5. thing is a base, so you put
6. the big fat end down and
7. the little end into the cap--
8. into the main tube

J: 9. Into the main tube?

[J is confused, he has no place to install the STAND]

A: 10. Yeah. Into the bottom of it.

J: 11. How do I get it into there?

2.4.3 Goal redundancy

Goal redundancy occurs when the requested goal (or subgoal) is already true. In some sense, it is a special kind of goal obstacle where the goal to be fulfilled is blocked because it is already satisfied. It is a simple goal obstacle because nothing has to be done to get around it. However, it can lead to confusion on the part of listeners because they may suspect they misunderstood what the speaker has requested since they wouldn't expect a reasonable speaker to request the performance of an already completed action. It provides a hint that miscommunication has occurred. In Lines 3 to 7 of Excerpt 17 [70] below, S discovers that U's request in Line 1 to add a roleset to the concept Person on the screen has already been satisfied. Lines 8 and 9 indicate that part of the cause of the confusion was that U's goal focus and S's goal focus (which was the goal focus that S believed U wanted) differed - i.e., U really meant to "change" the roleset instead of "adding" it to concept Person.

Excerpt 17 (Teletype with common graphics display)

U: 1. add roleset named Nickname to Person,
2. w/ number facet (0 NIL) and vr TEXT

S: 3. There appears to be an error in the
4. display... There is a roleset named
5. Nickname on Person, and it already
6. has a vr. Do you want me to
7. display it?

U: 8. Change the number and vr as indicated
9. and display please

2.5 Repairing miscommunication

The protocols in this chapter highlight numerous kinds of confusions that can lead to miscommunication. This section tries to briefly summarize the basic confusions and enumerate some methods around several of them that are suggested from analysis of the protocols. In particular, techniques for repairing reference confusion will be suggested. These repair techniques will motivate the knowledge sources and algorithms that will be developed in Chapters 5 and 6.

A problem that occurs during reference identification is finding, unexpectedly, more than one referent. The excerpts show that such ambiguity is often due to a speaker's underspecified description. The protocols suggest that a listener has several ways around this. He can ask the listener for clarification; he can search his knowledge of features and their values and consider their hierarchical relationships, dropping any imprecise feature values in the description; or he can attempt to reduce the set of referents down to one by trial and error (i.e., trying to see if a referent fits the speaker's current or future requests). A couple of those methods are clearly demonstrated in the excerpts.

Another confusion occurs when no referent is found during referent identification. One way this occurs in the excerpts is when the listener misfocuses. Detection of misfocus can only be determined by the listener looking back in the dialogue and his previous actions to see if there are hints of improper focus. The protocols show that misfocus often occurs after the speaker signals a problem in his utterance or when he abruptly changes the course of the dialogue. If such spots exist, then the listener can try performing differently than he had originally (e.g., shifting focus if he hadn't previously shifted there or vice versa). When no hint of misfocus exists, the listener's only other recourse is to try other objects around him to see if they would suffice as the referent.

The excerpts show that sometimes the confusion isn't due to finding too many or too few referents but is due to the inability of the listener to commence reference identification. This occurs because the description is too confused. Often, the confusion is due to the use of a bad analogy in a speaker's description. Analogies are sometimes too vague. In such cases, a listener must ask the speaker to clarify his description. Other times, the problem is that the analogy is too specific. In that case, the listener can try substituting less precise feature values in the speaker's description or simply drop the feature value that was too specific.

One surprising confusion that shows up in the protocols is the case where the referent is found too easily. The listener is given a description that is overspecified. It provides correct feature values that allow the listener to find the referent but, after the referent is discovered, there are still more feature values being provided by the speaker. This led the listener to doubt his original choice. In this case, the listener can either ask the speaker to confirm that the correct referent was identified, or ignore the excess features specified in the description.

Sometimes the listener finds a referent but the action requested by the speaker to perform on the referent fails. Often this occurs because the speaker's description of the action and its relation to the referent is underspecified. The listener can ask for clarification or try to find another referent and test whether or not the action succeeds on it.

Finally, one feature value in a speaker's description is found to occasionally influence another feature's value. This occurs if one feature is dominant over the other. In those cases, a listener needs a list of features and their values and an explanation of how they interact with each other.

2.6 Summary

I have attempted to show in the preceding sections that miscommunication occurs often in the real world between human conversants. It seems inevitable that miscommunication will also occur if people and computers cooperate on tasks, and so computers will have to be able to handle such problems. Miscommunication is <u>often</u> resolved subconsciously (possibly in a manner analogous to a relaxation process).

Many times, however, it can only be detected when the hearer's mental state doesn't agree with his perception of the physical world. In essence, then, there are two kinds of miscommunication – the easy ones that can be resolved instantly and the ones that are actively noticed and that require the hearer to step back and consider past dialogue or to ask for clarification from the speaker.

Miscommunication of goals, actions or plans is very hard for computer programs to deal with, at this time, because much more flexible representation schemes are needed. For example, it is hard to define the relaxation of an action. Hence, we need to develop a flexible way to deal with actions and their effects. There are, however, things that can be dealt with given our current technology. These have to do with the case of reference identification and possible reference failures.

3. REFERENCE IDENTIFICATION

3.1 Introduction

Reference is a way for participants in a conversation to discuss the same concept. People use words to refer to objects, places, ideas, and people that exist in the real world or in some imaginary world. These words include names (e.g., "Boston"), specific descriptions (e.g., "the large violet tube with two cylindrical outlets"), or more complex forms of reference such as reference by inference (e.g., "the thing that turns"). In this work my concern is with reference to the real world – extensional reference. My interest is primarily with descriptions of objects and how listeners go about determining which, if any, objects fit a speaker's description.

Reference identification is the actual process a listener goes through to determine what extensional or intensional element (i.e., the referent) is being described by a speaker. The process itself can entail a search of the listener's physical surroundings, a search of the listener's memory, inference on the part of the listener to get the speaker's description into a form that fits the listener's perspective of the world, or even the creation of the referent itself.[9]

The reader may wonder whether it is reasonable to consider reference identification as separate from the whole process of language understanding or whether they are too intimately tangled. There is evidence presented by Cohen [20, 22] that a speaker attempts as a separate step in his overall plan of communication to get a hearer to identify a referent. He provided grounds for an IDENTIFY action by illustrating particular requests to identify from his water pump protocols. For example, utterances like "Notice the two side outlets on the tube end" or "Find the rubber ring shaped like an O" showed that the speaker wanted the hearer to perform some kind of action. That action is the IDENTIFY act, which is to

[9] One could refer to a generic member of some class instead of any one particular element of that class. In that case, a representative of the generic could be created by the listener (in intensional form) for use in future references. For example, consider utterances like "The elephant is a large mammal" [69] or "Consider a pink elephant."

search the world for a referent for the speaker's description (and thus identify it). Cohen also showed that the hearer's response to a request to identify provided further evidence. He pointed out excerpts in the protocols where hearers responded to a request to identify with a confirmation that the identification had actually occurred (e.g., "Got it.").

I examined Cohen's protocols [20, 22] from a different perspective. He looked at them primarily from the speaker's viewpoint while I examined them from the hearer's. My analysis of the videotapes has shown that hearers often react to requests to identify in a very stylized way - unless something goes wrong. For spoken requests, they begin looking around the physical world in front of them for an object that fits the set of features that they are hearing in the speaker's description.[10] They pick up an object and examine it closer. Many times they choose a particular object as the referent before they hear the speaker's complete description. If later parts of the speaker's description contradict their original choice, they put their first choice aside and look for another.

This indicated to me that reference identification is a complex and ongoing task that involves more than a listener being handed a complete template for some object and being asked to find a match for it in the world. Reference identification appears to proceed in stages. The first stage is a cursory search of the physical world around the listener. The listener tries to find anything that fits the set of features he has heard so far. While the unfinished speaker's description is normally ambiguous, listener's often go ahead and noncommittally choose one of the set of possible referents until they hear information that contradicts their choice. Whether or not a choice is made this early depends on how small a set of possible referents is currently available. If the set is large, listeners in the protocols often waited for more information before grabbing for one of them. For example, if the speaker's description so far was "the red..." and there were several red objects, the listener often waited before taking one of them. Some listeners would pull all the red objects out of the set of objects and put them in front of them forming a group of referent candidates.

[10] Results of spoken requests can be decomposed because they usually come out slowly and piecemeal. It is harder, however, to tell what is happening in non-verbal requests (i.e., written or teletype) because the whole request is often instantly in front of the reader.

The second stage is one of actually making a firm choice. As evidence gets stronger (such as when all but one element is ruled out), the listener would often physically take the object, either holding it or putting it in front of him. At this point, the third (and normally the last) stage begins. This stage is the confirmation stage. Here the listener tries to confirm that he has made the proper choice. If there are still unheard portions of the speaker's description left, the listener continues to examine his selection to confirm that it fits the rest of the speaker's description. When the speaker's description is finished, some listeners would pause and examine the object closely, possibly reviewing each part of the speaker's description at that time. Other times the listener would try to see if the selected object fit with subassemblies created in a previous action, or, if the speaker explicitly specified a particular action to perform with the object, try to perform the requested action on the object. Failure of the confirmation stage leads to a fourth stage - the retry stage. This stage requires trying again to find a referent that works. The listener checks over previous choices that may have been made in Stage 1 to see if any of them work better. This stage often results in a listener finding a referent but occasionally leads to requests for clarification.

The protocols are especially revealing in the cases when things went wrong. They show that listeners can change their mind, dropping one choice and attempting to find another; that they can tolerate certain levels of imprecision or mistakes; and that they can often determine when they are lost and need more information or help from the speaker.

The rest of this chapter describes previous natural language systems and their attempts at formalizing the referent identification task.

3.2 Previous computational paradigms

This section describes three examples of natural language systems that have been developed and the reference mechanisms that are part of them. All fit into the same basic paradigm: put the speaker's description into a searchable form (i.e., parse and semantically interpret the speaker's description) and then use that form as a pattern that can be compared against objects (i.e., the possible referents) in the world. A referent is found when a match occurs between the pattern and one or more of the objects. The pattern and a target referent match each other if all the attributes specified in the pattern exactly fit the corresponding attributes in the target. There is variability in each of the reference schemes described below in what pattern is generated, how the world is represented, and how the actual search progresses, but the general scheme remains the same. Success in all cases occurs if and only if a perfect match exists between all the pattern's attributes and the corresponding attributes on a target.

3.2.1 Reference in SHRDLU

A program called SHRDLU, completed in 1971 and written by Terry Winograd at MIT, works on a small data base describing a world of geometric solids such as rectangular blocks and pyramids [81, 82]. SHRDLU can display this "micro-world" on a CRT screen and actually simulate the movement of elements of that world with an "imaginary" robot arm. The user can request SHRDLU to perform certain manipulations of the blocks and to answer questions about the current scene. In addition SHRDLU can comprehend declarative sentences (e.g., "The blue pyramid is nice.") and imperative sentences (e.g., "Pick up a big red block.") as well as procedural statements (e.g., "A steeple is a stack which contains two green cubes and a pyramid." Here "steeple" is defined procedurally because the goal of a "steeple" requires first finding a "stack" and checking that it contains two green cubes and a pyramid.)

SHRDLU consists of a set of recursive procedures that can profitably describe natural language grammars and parsers. It uses a fairly comprehensive grammar of English; its parser is organized around syntactic units, which play a primary role in determining meaning; and for each syntactic unit, there exists a program (written in the language PROGRAMMAR also developed by Winograd at MIT) which operates on the input string to see if it can represent that type of unit. In the process of doing this, it calls on other syntactic programs (and even possibly recursively on itself). These programs incorporate descriptions of the possible orderings of words and other units. When the parser finds a syntactically acceptable phrase, it performs a semantic analysis on it to determine whether to continue along the current line of parsing.

Winograd's system is based on a theory by Halliday [33] called Systemic Grammar which incorporates both syntactic and semantic information. It describes the interaction and dependency of different features on each other and is concerned with the way language is organized into units, each of which has a special role in conveying meaning. Systemic grammar uses the WORD as its basic building block. Classes of words such as "noun", "verb", and "adjective" are used.

The next unit above WORD is GROUP. Such groups include noun groups, which describe objects; verb groups, which convey messages about time and modality; prepositional groups, which describe simple relationships; and adjective groups which convey other types of relationships and descriptions of objects [83]. Each of the groups has "slots" for the words of which it is composed. For example, a noun group has slots for the "determiner", "numbers", "adjectives", "classifiers", and a "noun."

The most complex unit of the language is the CLAUSE. It is used to express relationships and events that involve time, place, and manner. A clause can be a QUESTION, a DECLARATIVE, or an IMPERATIVE; it can be in the "passive" or "active" form; it can be a YES-NO or WH-question, and so forth. Clauses can be made up of other clauses and they can be used as parts of groups in many ways.

The interpretation of a request by SHRDLU is done by making use of a detailed world model that describes the current state of the blocks and its knowledge of procedures that allow it to change state. The model is a symbolic representation that shows those aspects of the world that are relevant to the operations needed to discuss it. The model is represented in a system called PLANNER [36, 81]. PLANNER is a superset of LISP that:

- Can automatically traverse tree structures depth first;
- Provides facilities for automatic backup (e.g., backing up a tree);
- Provides built-in pattern matching;
- Supports a data base with functions for updating, adding, and deleting information; and
- Provides procedural knowledge.

The PLANNER data base is a collection of data items (or "facts") that have been asserted. For example, (IS B1 BLOCK) and (DIMENSION-OF B1 (10 20 30)) are typical data base entries that can be asserted. All PLANNER functions return either SUCCESS or FAILURE. The data base is searched by using the PLANNER function "THGOAL." THGOAL allows one to assert what conditions should be true to satisfy the request. In using THGOAL, however, it is not necessary to be specific, one can use patterns in the search specification.

A PLANNER data base of part of the water pump world might contain assertions such as

```
(IS CYLINDER PHYSICAL-OBJECT)
(IS TUBE CYLINDER)
(IS TUBE FUNCTIONAL-OBJECT)
(IS VIOLET COLOR)
(IS BLUE COLOR)
(IS LARGE SIZE)
(IS SMALL SIZE)

(IS MAINTUBE TUBE)
(COLOR MAINTUBE VIOLET)
(SIZE MAINTUBE LARGE)
(IS OUTLET1 OUTLET)
(IS OUTLET2 OUTLET)
(SUBPART MAINTUBE OUTLET1)
(SUBPART MAINTUBE OUTLET2)
(IS OUTLET1 CYLINDER)
(IS OUTLET2 CYLINDER)
```

The dictionary definitions of words in SHRDLU were written in PLANNER. They provided both grammatical information useful in parsing an utterance and contained templates of PLANNER assertions that represented the "meaning" of the word. They look something like the examples below. The first entry in each definition is the word itself. The second entry contains the grammatical category of the word, the kind of thing represented by the word, and then a list of PLANNER templates.

```
(CYLINDER
   ((NOUN (PHYSICAL-OBJECT
           ((MANIPULABLE CYLINDRICAL) NIL)))))

(TUBE
   ((NOUN (PHYSICAL-OBJECT
           ((MANIPULABLE CYLINDRICAL)
                         ((IS ? CYLINDER)
                          (IS ? FUNCTIONAL-OBJECT)))))))

(OUTLET
   ((NOUN (PHYSICAL-OBJECT
           (NIL ((IS ? OUTLET)
                 (IS ? FUNCTIONAL-OBJECT)))))))
```

```
(VIOLET
   ((NOUN (PHYSICAL-PROPERTY
          (NIL ((IS ? COLOR))))))
```

Winograd's SHRDLU system tries to semantically interpret what each sentence means by generating PLANNER programs for each word. Semantic interpretation proceeds by inspecting both the syntactic structures and the meaning of each word and using them to build up "theorems" that can be used by PLANNER to perform actions or to answer questions, or for the syntactic system itself to decide if a proposed noun group makes sense.

In this paradigm, reference identification is performed by creating a PLANNER program that describes the object whose referent is wanted, and then asserting that the description be "True." For example, the phrase "a large violet tube with two cylindrical outlets" would be represented by the following PLANNER program.

```
(THPROG(X1)
   (THGOAL (#IS $?X1 #CYLINDER))
   (THGOAL (#IS $?X1 #FUNCTIONAL-OBJECT))
   (THGOAL (#COLOR $?X1 #VIOLET))
   (THGOAL (#SIZE $?X1 #LARGE))
   (THFIND 2 $?X2 (X2)
      (THGOAL (#IS $?X2 #CYLINDER))
      (THGOAL (#IS $?X2 #OUTLET))
      (THGOAL (#SUBPART $?X1 $?X2))))
```

A search for a referent is done by asserting the above statement and then seeing if it succeeds. The assertion causes PLANNER to search the data base to try to find an object, $?X1, that satisfies each of the specified goals (i.e., each THGOAL statement and the embedded THFIND statement). Whenever one of the goals fails, the system can back up and try another match for $?X1 or $?X2 to see if it can succeed in satisfying all the specified goals. If it succeeds, then a referent has been found; otherwise, no referent has been found and the search fails. In the case of the data base given earlier, this would result in PLANNER doing the match with $?X1/MAINTUBE and $?X2/{OUTLET1 OUTLET2}. Note that if one of the goals in the PLANNER request is incorrectly specified, the reference mechanism will fail (or, accidentally, discover another data base element that it incorrectly assumes is the proper referent). SHRDLU, thus, must assume that a user's input is perfect if it is to work properly.

3.2.2 Reference in LUNAR

Woods [84] provides a way for expressing in a formal language based on predicate calculus the meaning of a sentence. A derivative of this scheme - called the Meaning Representation Language, or MRL - was used in building the Lunar Sciences Natural Language Information System which contains information about samples of lunar rocks and soils that were returned by the Apollo moon missions. The system, called LUNAR, was developed by Woods and his co-workers at Bolt Beranek and Newman Inc. [86]. It is an experimental question answering system that was designed to help geologists access, compare, and evaluate the data. It is able to accept grammatically complex sentences, involving nested dependent clauses, comparative and superlative adjective forms and some types of anaphoric reference. LUNAR performed well in its domain of geology (e.g., in a demonstration of LUNAR in 1971, 78% of the questions asked to the system were understood and answered correctly [87]).

The syntactic component of LUNAR is an augmented transition network grammar. The grammar is implemented by an augmented transition network, or ATN [85]. An ATN is a generalization of phrase structure grammars that has recursion, tests and actions; as well as the power of a finite state automaton. An ATN is implemented as a set of recursive procedures that can efficiently describe natural language grammars and parsers. It consists of sets of nodes and branches emanating from the nodes. Each branch is a labeled directed arc that is allowed to specify a condition and a sequence of actions to be taken if the condition is met. ATNs enable one to try out different parsing strategies on variably large phrases in a sentence, to store information relating to the success of those strategies as they are being carried out, and to recognize whenever a given strategy has failed so that a new strategy can be tried. In particular, ATN parsers employ a depth first backtracking algorithm as they attempt to traverse a path of nodes and arcs leading from the starting state to some accepting state. The value of the input string and the tests applied to it determine which paths are taken in the ATN. A sample ATN for parsing noun phrases is shown in Figure 3-1.

LUNAR's ATN parser attempts to map an input request into a deep structure representation. As transitions occur in the nets, the parser builds up parts of a deep structure tree and stores them in "registers" (using the SETR command), until they

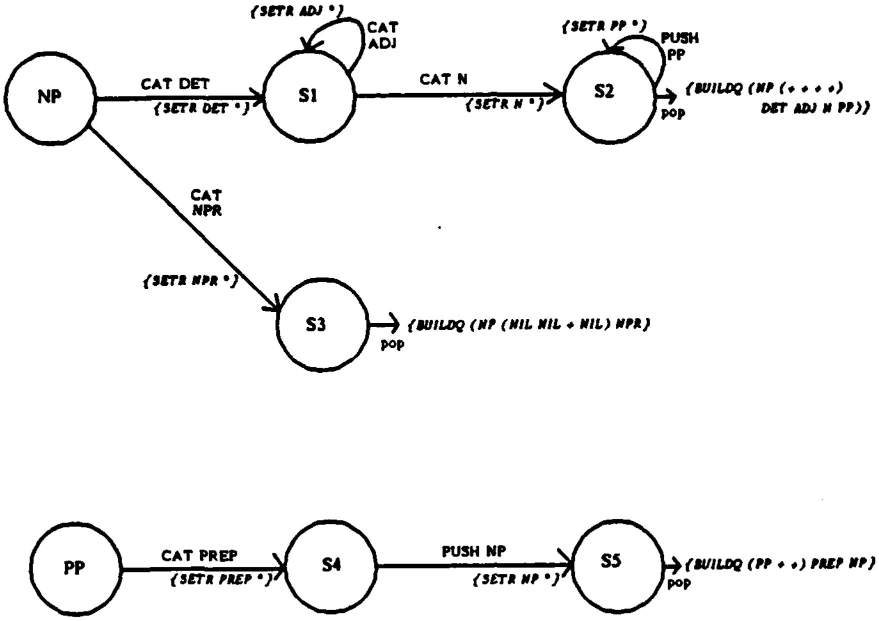


Figure 3-1: An ATN for Noun Phrases

can be combined into larger groups (using the BUILDQ command), and, ultimately, into a complete representation of the input. LUNAR defines a sentence as consisting of a subject noun phrase; an auxiliary verb component that specifies the tense, modality, and aspect of the sentence; a verb phrase containing the main verb; the direct and indirect objects; and possible adverbial and prepositional phrase modifiers [86]. The basic approach of the parser is as follows: at the sentence level, it tries to determine whether the input string is declarative (e.g., "John needs money.") or an interrogative (e.g., "Does John need money?"). At the next lower level, the parser attempts to find the subject noun phrase, the verb phrase, and so on. Each attempt at parsing those constituents requires descending into lower levels looking for such forms as adjectives, determiners, prepositions and the like. In the process it is possible for recursive calls to be made to some of the nets.

Consider the description "the large violet tube with two cylindrical outlets." The parse generated using a piece of the LUNAR grammar like that in Figure 3-1 would look something like the one shown in Figure 3-2.

Semantic interpretation in LUNAR involves translating the parse of the sentence

```
NP  DET the
    ADJ large
    ADJ violet
    N   tube
    NU  SG
    PP  PREP with
        NP  DET NIL
            ADJ cylindrical
            N   outlet
            NU  PL two
```

Figure 3-2: Sample Parse of a Noun Phrase

into a program in MRL that can be executed to retrieve or compute the answer. MRL is essentially a retrieval program that computes the truth values of propositions or carries out commands. It consists of primitive commands, functions, and predicates which may be combined and quantified [84, 87]. The basic form of an MRL query is:

(FOR <quant> X / <class> : (p X); (q X))

where

- <quant> is a quantifier like EVERY, SOME, TWO, and so on,
- X is the variable that is being quantified over,
- <class> is the domain of the quantification (i.e., the set over which X can range), such as TUBE, CYLINDER, VALVE and so forth,
- (p X) is a predicate that can be used to restrict the domain of quantification (e.g., (PART-OF X MAINTUBE)), and
- (q X) is the expression being quantified (which is either a predicate such as (COLOR X VIOLET) or an action such as (PRINT X)).

The actual interpretation of a sentence into a query occurs in two phases. The first phase looks to see if there are any operators or commands such as NOT, TEST, and so forth, that govern the sentence. This phase is performed before actual examination of the input sentence itself (and is thus a preprocessing phase). The first phase consists of a search for rules which match anything in the input. The handling of compound sentences, declarative sentences, imperative sentences, and questions begins here. The second phase uses the main verb in the sentence and rules associated with the verb to interpret more of the sentence.

The semantic interpretation of a sentence normally requires interpreting one or more noun phrases. An important aspect of the meaning of each noun phrase is the notion captured by quantifiers such as "all," "every," or "three." One of the first tasks of semantic interpretation of noun phrases, hence, is the examination of the determiner structure of the noun phrase to decide what kind of quantifier should govern it. This quantifier structure is used during the rest of the analysis when the noun of the phrase and relative clauses are handled. LUNAR treats some noun phrases as special cases. In particular, topic descriptions are handled by a special set of rules used to translate their syntax trees into Boolean combinations of important phrases.

Semantic interpretation rules are used to map the parse into MRL. They consist of patterns and actions (patterns that determine if a rule applies and actions that specify how to construct the semantic interpretation). The pattern describes semantic conditions that must hold. It is composed of numbers that denote a position in a template of some syntactic constituent, Boolean operators, and predicates that check if a particular condition holds. Each rule can fire other rules in the process of determining whether or not they are satisfied. The set of templates of syntactic constituents below are ones that can be used to cover some noun phrases.

```
NP.N=NP  N (1)  (noun of a noun phrase)

NP.DET=NP  DET (1)  (determiner of a noun phrase)
           NU  (2)  (number of a noun phrase)

NP.ADJ=NP  ADJ (2)  (adjective modifying a noun phrase)

NP.ADJ-ADJ=NP  ADJ (1)  (adjectives satisfying a
               ADJ (2)       noun phrase)

NP.PP=NP  N PP PREP  (1)  (preposition and
               NP    (2)      object modifying a noun phrase)
```

The target part of the rule defines the actual semantic interpretation. It is composed of fragments of MRL that define conditions that must be satisfied by the element being described. These fragments can be collected to form an MRL function capable of determining the referent of the user's description.

The pattern pieces of LUNAR semantic interpretation rules capable of interpreting the parse in Figure 3-2, and their associated MRL fragments are shown below. The predicate MEM checks to see if the constituent is semantically marked as indicated.

EQU checks for equality. The numbers refer to positions in the set of templates of syntactic constituents mentioned above. *PP* and DLT are place holders to tie the MRL for one constituent into an MRL generated for another constituent.

```
(N:TUBE
  ((NP.N  (MEM 1 (TUBE)))
   (OR (NP.ADJ  (MEM 1 (SIZE)))
       (NP.ADJ  (MEM 1 (COLOR)))
       (OR (NP.ADJ-ADJ  (AND (MEM 1 (SIZE)) (MEM 2 (COLOR))))
           (NP.ADJ-ADJ  (AND (MEM 1 (COLOR)) (MEM 2 (SIZE))))
       (NP.PP (AND (EQU 1 WITH) (MEM 2 (OUTLET))))))
    —>
       (for the y/SUB-PART:  (AND (COLOR y COLOR-VAL)
                                  (SIZE y SIZE-VAL)
                                  (FUNCTION y TUBE)
                                  *PP*); T; DLT)

(N:OUTLET
  ((NP.N  (MEM 1 (OUTLET)))
   (NP.DET  (NP.DET.INTEGER T))
   (NP.ADJ (MEM 1 (SHAPE)))))
   —>
      (for NUMBER-VAL x/SUB-PART:  (AND (SHAPE x SHAPE-VAL)
                                        (FUNCTION x OUTLET)
                                        (PART-OF x
                                                MAIN-PART)); T;DLT)
```

The MRL interpretation of N:OUTLET plugs into the *PP* slot in the interpretation of N:TUBE.

The complete semantic interpretation of the parse would yield something like:

```
(for the y/SUB-PART:  (AND (COLOR y VIOLET)
                           (SIZE y LARGE)
                           (FUNCTION y TUBE)
                           (for 2 x/SUB-PART:
                             (AND (SHAPE x CYLINDRICAL)
                                  (FUNCTION x OUTLET)
                                  (PART-OF x y)); T)); T)
```

A referent, such as the referent for the example shown above, is found by executing the MRL query. The system searches the data base to look for exactly one entry that satisfies all the conditions specified in the MRL request. Figure 3-2 provides a sample of a typical data base. In that data base, SUB-PART S0001 satisfies all the conditions specified in the MRL. Notice, however, that if one of the conditions had been incorrectly specified, the MRL query would have failed to find anything to satisfy the request. LUNAR, hence, must always assume that the user's request is perfect.

SUB-PART	FUNCTION	SHAPE	COLOR	SIZE	PART-OF	...
S0001	TUBE	CYLINDRICAL	VIOLET	LARGE	NIL	
S0002	OUTLET	CYLINDRICAL	VIOLET	SMALL	S0001	
S0003	OUTLET	CYLINDRICAL	VIOLET	SMALL	S0001	
S0004	CAP	CYLINDRICAL	BLUE	LARGE	NIL	
S0005	VALVE	ROUND	PINK	SMALL	NIL	
⋮						

Figure 3-3: Sample data base entries

As I have shown in my description of LUNAR, an important part of reference identification is the semantics used to represent the description whose referent is sought. In both LUNAR and SHRDLU, procedural semantics [84] is used.[11] Procedural semantics represents the semantics of a set of elements by a procedure that can be directly executed to recognize members of the set. In SHRDLU, a set is described as a PLANNER theorem that, when executed, exhaustively (if the data base is finite) searches the data base for elements that satisfy the theorem. LUNAR's MRL works similarly. Both find a referent by trying to enumerate those elements in their data base that satisfies a set of conditions about the referent. One flaw in both these schemes is that they must search the entire data base to enumerate a set.[12] The work on focus by Grosz [30] described in the next section shows a more efficient way to search for referents. Another problem with both schemes, which I pointed out earlier, is that they require that the conditions about the referent expressed in the speaker's description be correctly specified. If they aren't, neither method can find the proper referent.

[11] MRL, however, is more expressive than SHRDLU's PLANNER representation of a description because it is much closer to first-order predicate calculus.

[12] LUNAR does provide special enumeration functions that can make this more efficient for a select number of sets.

3.2.3 Reference in TDUS

TDUS is a natural language system developed at SRI International [61, 63]. It is capable of handling natural language dialogues about an ongoing mechanical assembly task. It expanded the work in natural language understanding beyond question answering and story understanding to extended dialogues. TDUS has the ability to follow a task as it progresses and shift the context of the dialogue in unison. This system (and its predecessors at SRI [30, 77]) introduced the notion of discourse knowledge [30] as an essential part of language understanding. It used information about the specific task (the assembly of an air compressor) and the goals of the participants in the dialogue.

Since knowledge about the task domain was used in TDUS, a way was needed to encode that knowledge. For example, an utterance like "undo the last piece" shows how one needs to represent and use knowledge about the current and previous states of the task. A representational formalism was developed [35] that allowed the representation of the changing environment of a task. This formalism was based on partitioned semantic networks [34, 35]. It allows a hierarchical decomposition of knowledge. Actions, events and objects could all be represented in the network.

Knowledge about the dialogue context was also used in TDUS. The SRI work showed that a speaker's utterances are affected by both the task domain and the context of the dialogue itself. For example, listeners use the context of previous utterances when interpreting the current one. Two important aspects of dialogue context are the focus [30] and goals [18, 3, 69]. Focus is a means of selective attention of currently relevant parts of the dialogue and elements in the real world. It changes dynamically over the course of a dialogue. A speaker's utterance helps guide the listener in determining the current focus as well as knowledge about the task itself. Focus is crucial for performing reference identification, especially when interpreting anaphoric definite noun phrases. Goals have to do with the task domain, the dialogue participants, and social conventions. In TDUS, goals about the task domain and some goals about the knowledge of dialogue participants were considered.

TDUS is built around a system called DIAMOND [54]. DIAMOND provides a framework for defining the language that can be used in TDUS. It is a programming language that allows a programmer to define (syntactic) phrase structure rules and semantic interpretation rules. A sample rule for noun phrases is shown in Figure 3-4 (pp. 16-17, [61]). The complete grammar - called DIAGRAM - is described in [62].

Stage 1

```
TEMPLATE
 NP = {DET/QUANT} (ADJ) NOUN (PP);

CONSTRUCTOR
 (PROGN (@FROM NOUN NUMBER)
        (@FROM DET DEF)
        (COND ((@ ADJ) (OR (AGREE TYPE ADJ NOUN)
                           (F.REJECT 'NO-AGREEMENT)))))
```

Stage 2

```
TRANSLATOR
 (@SET SEMANTICS (COMBINE (@ SEMANTICS ADJ)
                          (@ SEMANTICS NOUN)))
```

Stage 3

```
INTEGRATOR
 (@SET D.IDENT (RESOLVE (@ SEMANTICS)))
```

Figure 3-4: A sample DIAMOND rule for noun phrases

The NP definition provides a template for the sample noun phrases. Phrases such as "the violet tube," "one tube," and "the violet tube with the two outlets" all match the above NP template. The CONSTRUCTOR part of the rule is executed when the NP template is matched. It assigns attributes (such as copying the value of the DEF attribute from the DET constituent to the noun phrase being built) and checks to make sure that the attributes are consistent.

The TRANSLATOR part of the rule is used when the entire utterance containing the noun phrase has been parsed. It considers how the constituent fits into the whole utterance. Its rules are used to map words and phrases to forms in the model of the domain represented in the partitioned network [35]. The rules can also be used to reject phrases because they do not make sense after considering domain knowledge. The rules create fragments of the network to correspond to phrases that are meaningful.

The INTEGRATOR is used to relate parts of a phrase with actual domain elements.

This is the stage where reference identification occurs. Here, TDUS departs sharply from previous work by tightening the paradigm provided by Winograd, Woods and others. A complex control strategy is utilized that takes into account focus, goals, domain knowledge, and dialogue knowledge when searching for a referent.

3.2.3.1 Focus and reference in TDUS

The work on focus by Grosz [30, 32] provides a better way to resolve referents by constraining the search space. For definite noun phrases, the choice of possible referent candidates is guided by the focus mechanism. The information provided in the definite noun phrase itself (i.e., by the head noun and any modifiers) is used to distinguish the referent from other objects in focus. Grosz showed how both the surrounding non-linguistic environment and the global linguistic context of preceding discourse are part of focus and how it is used to resolve definite noun phrases.[13]

Focus changes as a dialogue progresses and the participants change their focus of attention in the world (this is referred to as a focus shift). The elements in the knowledge base that are currently relevant are highlighted by partitioning them into a unit called a focus space [30]. After a shift in focus, the new focus space becomes active. There is only one active focus space at a time. The previous focus space can become open (i.e., inactive but left in an unfinished state so it may eventually become active again) or closed (i.e., inactive and no longer relevant) [30]. Open focus spaces and the current active focus space can be related to each other in a hierarchical fashion. They are used to represent elements in explicit focus, i.e., elements explicitly discussed in preceding discourse. An element could also be implicitly in focus. Such elements are related to the element that is explicitly in focus and become implicitly in focus because of that relationship. A shift in focus in task dialogues is strongly related to the task itself since the dialogue often parallels the task's structure (e.g., when a new task is begun or an old one finished, a shift occurs). Linguistic cues also provide a way to shift focus [30, 57, 69, 58]. A speaker could shift focus directly by saying that the current discussion is completed and that a new one is to begin (e.g., "I'm finished. What's next?" [30]) or more subtlely with linguistic clues that suggest

[13] See Sidner [69] for a description on the use of focus to resolve anaphoric definite noun phrases. Webber [78] provides a formal treatment on the handling of anaphoric references.

a shift (e.g., "Okay. Now...," "But anyway..." [57]). Focus shifts can also be suggested in the use of a definite noun phrase. If the definite noun phrase refers to an element in either the active focus space or in an open focus space, then no focus shift occurs because the referent is right there. A definite noun phrase reference, however, to a subtask or a new task will cause a shift in focus. More detailed criteria for shifting focus can be found in [57, 69, 58].

Focus has been represented in the partitioned semantic network and used to help guide the search for referents of noun phrases [30]. The focus is computed dynamically as the dialogue progresses, highlighting different (and currently relevant) parts of the network. Figure 3-5 provides an example of a partitioned network. T1 and T2 are two tubes that exist in the world. The FS1 box drawn around T1 represents the current focus space. A search for a tube would start first with T1 (and not with T2) because it is currently in focus.

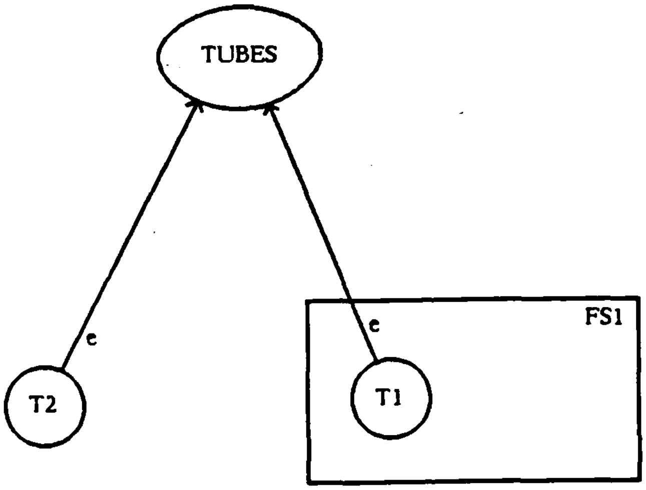


Figure 3-5: A partitioned semantic network

As I mentioned above, Grosz distinguishes between two kinds of focus - explicit focus and implicit focus. Explicit focus is the relevant part of the knowledge network that was explicitly mentioned in preceding utterances. Related to the task elements that are in explicit focus are elements that are closely tied to them. These elements are in implicit focus. They include such things as the subparts of objects in focus, subactions or objects of the task in focus, or an element that is evoked through some inference from the object in explicit focus [78]. For example, if "the desk" is explicitly in focus, then "the top drawer" is implicitly in focus. This distinction between explicit and implicit focus is important during referent identification because often the referent of a definite noun phrase is in implicit and not in explicit focus.

The focus mechanism provides a scheme for resolving definite noun phrases. The search for the referent of a definite noun phrase can begin by examining the objects currently in focus. The modifiers and head of the definite noun phrase can be compared to the description of each object. If a match occurs, then a referent has been found; otherwise, those objects implicitly in focus (e.g., subparts of an object in focus or associated objects) can be examined in turn for a match. The actual implementation is done by dividing the partitioned semantic network into two pieces, the QVISTA and the KVISTA. The QVISTA contains a representation of the object described in the speaker's noun phrase (this representation was produced by the CONSTRUCTOR and TRANSLATOR stages of analysis). The KVISTA represents all the relevant knowledge over which a match is to be considered. An initial version of it is given to TDUS but, as utterances are interpreted, the focus mechanism partitions the KVISTA into (overlapping) sections that represent the focus of attention of the dialogue participants (i.e., the focus spaces). When a match is found between the element in the QVISTA and a piece of the KVISTA, then the referent is found. The actual matching process is described in [23].

Figure 3-6 provides an example of a QVISTA that describes the noun phrase "the large violet tube with two cylindrical outlets." T1 is the node that represents the tube. The nodes CT1 ("Color of T1"), LT1 ("Relative Size of T1"), and S1T1 ("Subpart of T1") represent the modifiers used to describe tube T1. The nodes OUTLET1 and OUTLET2 represent the two subparts of T1. Nodes SHO1 ("Shape of O1") and SHO2 ("Shape of O2") denote modifiers of "outlets" in the prepositional phrase.

There are some problems with the focus mechanism in TDUS. First, the mechanism does not allow for backtracking after a focus shift occurs. This means that should a new utterance affect the shift or clarify ambiguity that occurred at the time of the shift, the system would not be capable of correcting for the mistake. Second, Grosz states (pp. 96-7, [30]) that if a referent cannot be found in focus

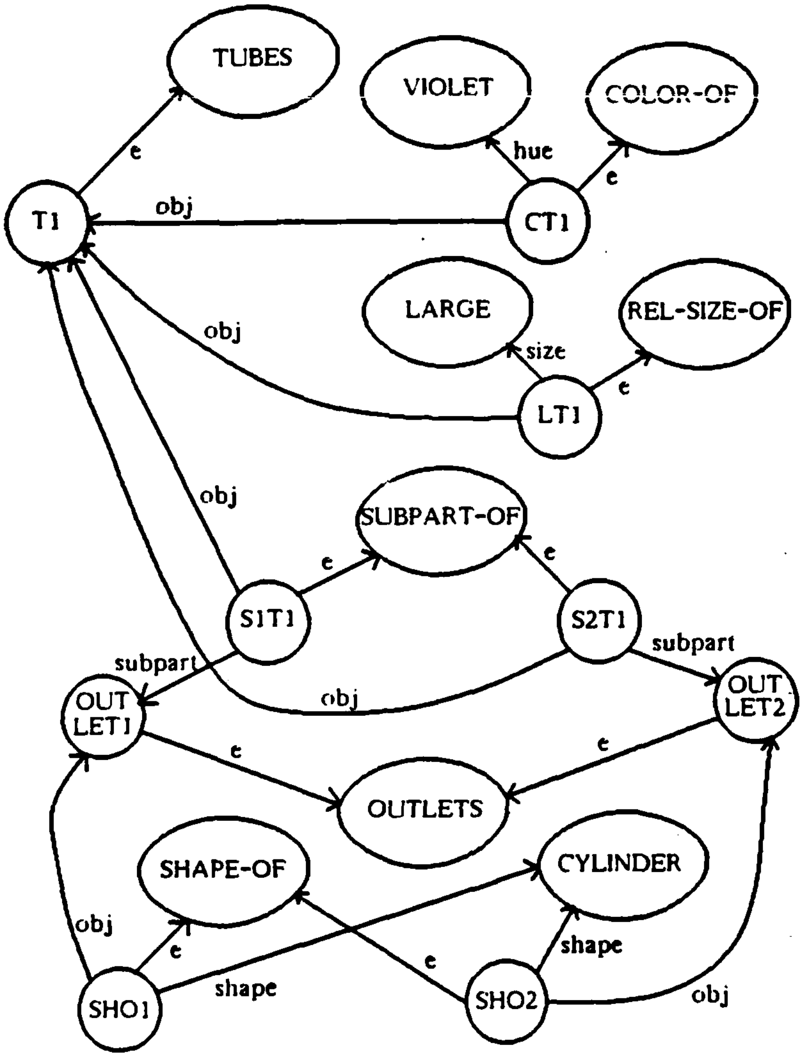


Figure 3-6: A sample QVISTA

(either in explicit or implicit), then modifiers in the definite noun phrase can be removed until a match does occur. This assumes that the problem is always due to a modifier. Many of the excerpts presented in Chapter 2 (e.g., Excerpt 7) showed that this is just not the case. Finally, since TDUS has no plan recognition, the focus mechanism is unable to recognize plans (i.e., it is limited to only one possible plan, so it is only necessary to determine the steps of the plan) and simply handles referent identification. This leads to trouble when new entities are created by actions. For example, this would mean that subassemblies generated during an assembly process would not have new names. Therefore, dialogues like the one in Figure 3-7 do not occur.

> S: "Take the flour and water and
> mix them together.
>
> Okay. Now take the dough and..."

Figure 3-7: A task dialogue where new elements are created

3.3 Summary

This chapter attempted to define the computational approaches to natural language understanding used by three successful systems. Each system differed (sometimes dramatically) in the way it represented knowledge about linguistics and the physical world and with the kind of parser and semantic interpreter that were employed. All of them, however, followed the same line of reasoning when it came to identifying a referent. A knowledge base was searched to see if a match could be found between the user's input and some element in the knowledge base. If no match was found, each system would give up the search with failure.

4. REFERENCE IDENTIFICATION IN FWIM PARADIGM

The previous chapter described three natural language systems. Each of them performed reference identification, using the same basic computational scheme, with a search for something that satisfies the speaker's uttered expression. In Chapter 1, I called their scheme the traditional approach and introduced a new approach, called FWIM. The FWIM approach rests on the claim that the reference understanding process does not follow a "find/didn't find" paradigm. In fact, the data presented in Chapter 2 support the new paradigm – communication is much more robust than the traditional approach suggests and people often recover from mistakes. This chapter describes the basic referent identification module used in the BBN natural language system. The relaxation component of the reference mechanism is described in Chapters 5 and 6.

4.1 The representational system

The representational framework of the system is a critical component of the BBN natural language system. Much of the power and robustness of the system comes from the richness and expressiveness of its knowledge representation system. The system uses the knowledge representation language KL–One which can represent general conceptual information using structured inheritance networks [8].[14] KL–One differs from previous representational systems because it provides a clean semantics that defines the inheritance of structured descriptions independent of a particular domain, taxonomic classification of generic knowledge, roles that describe functional relationships between concepts, and a way to attach procedures that can be invoked automatically. KL–One is used to construct knowledge bases of information that correspond to one person's beliefs about the world.

KL–One actually is built out of two sublanguages – a description language and an assertion language. The description language is used to build definitions of

[14] A more comprehensive description of KL–One can be found in [8, 9, 10]. KL–One is being superseded by a new implementation called KL–Two which is currently under construction [72, 76].

general terms or to construct individual instantiations of those definitions using other description terms and a small set of primitive operators. The assertion language is used to assert information about the world using elements in the description language. The assertions include statements that two descriptions corefer in a particular context or on the existence and identity of an individual in a particular context.

4.1.1 Concepts, roles and the taxonomy

KL-One descriptions are composed principally of one element, **Concepts**, which is itself divided into two types, Generic and Individual. Generic concepts are used to describe general terms that define a set of potential elements in the world. They are arranged in the inheritance structure to express generic knowledge in a taxonomic fashion. Individual concepts can be formed by using a generic concept as a template. An individual concept represents one individual in the world. For example, a knowledge base could contain generic concepts such as physical object, animate object, cylinder, tube, and human and individual concepts such as Bill (a human) or Tube#5 (a tube).

As I stated above, KL-One provides structured inheritance, and this is realized in a KL-One concept. A concept is defined by (1) combining the definitions of those concepts more general than it (the **SuperConcepts**), (2) using local information expressed in **Roles** attached to the concept and (3) using **Structural Descriptions** which define relationships between roles. A role describes possible functional relationships between concepts (e.g., the properties or the parts of the concept). A structured description can relate one role to another role by defining the relationship (e.g., it could state that two roles are identical or that one is included in the other). Figure 4-1 shows a sample of generic concepts arranged in a taxonomy to show subsumption relationships between concepts.

A concept is represented in the figure by an ellipse labeled with a **name**. One generic concept (the subsumer) is said to subsume another generic concept (the subsumee) if it is more general than the latter concept. This is represented in the figure by having the more specific concept "below" the more general one. An arrow in the figure (called a "SuperC link") points from the more specific concept to the more general one. ANIMAL, thus, is more specific than THING and HUMAN is more specific

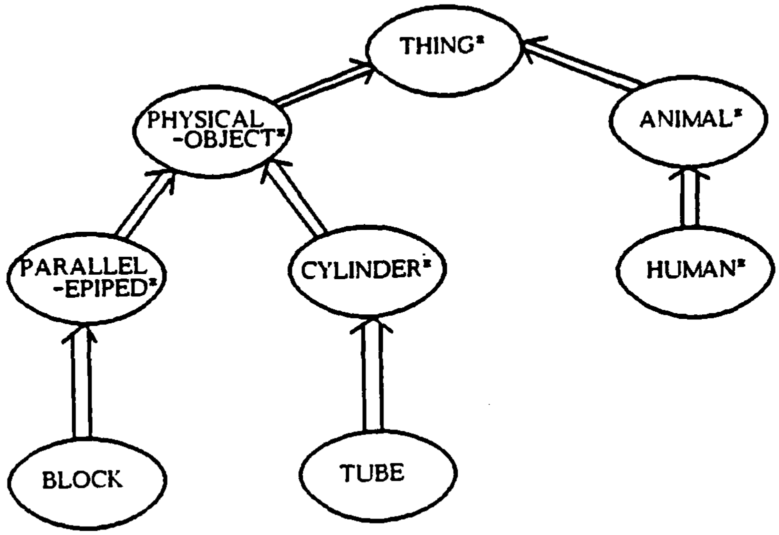


Figure 4-1: A KL-One taxonomy of generic concepts

than ANIMAL. Subsumption relationships are transitive, so THING also subsumes HUMAN. Another point to note in the figure is that a KL-One taxonomy always has a single root concept - normally called THING. THING subsumes all other concepts in the taxonomy. Those concepts in the figure marked with a "*" are said to be primitive concepts. A primitive concept is one that is not fully defined. Anyone using one of them must take that into account.

A role in KL-One provides general attribute descriptions about a concept. It defines functional relationships between the concept and other concepts, behaving like a two-place predicate. It is defined on a generic concept by a **RoleSet**. A roleset on a generic concept describes the set of intensional elements determined by that role (e.g., "subpart of an object"). Each individual instantiation of the concept will have a set of intensional elements corresponding to those defined for the role on the concept (e.g., "the end of the tube"). A role has its own structure with descriptions of its potential fillers (called its Value Restriction or V/R), its name (which is present for convenience but is not actually used by the system), and its number restriction (which expresses cardinality information about the number of possible fillers). A role and all of its structural information is inherited by all subconcepts of the concept to which the role belongs. Figure 4-2 provides an example of a roleset "Subpart" defined on the concept "Physical-Object." The roleset states that its name is "Subpart," that it

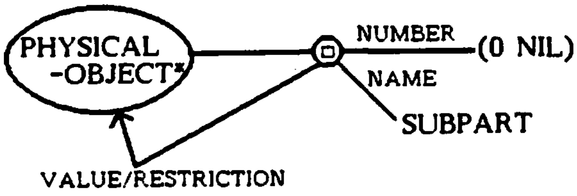


Figure 4-2: A RoleSet

can be filled by zero or more elements (its number restriction), and that the value of its fillers are restricted to be Physical-Objects. A roleset on an individual concept, called an **IRole**, defines the set of individual intensions for that concept and only that concept. They are used to represent a particular connection of a role to an individual concept (e.g., "the end of TUBE#5").

A roleset on a concept can appear on a subconcept below that concept. The lower roleset, however, can be modified by one of four relationships.

- **restriction**: the filler of the V/R can be restricted to a more specific form of the filler of the superconcept's V/R (this is also called "modification" and is represented by a "**Mods**" link). For example, a particular kind of Physical-Object is restricted to have exactly two **Subparts**, all which are CYLINDERs.
- **differentiation**: the role on a superconcept is divided up into subroles using a "**Diffs**" link. For example, the role **Subpart** on the concept Physical-Object could be differentiated into several subroles, such as **Engine**, **DriveShaft**, and **Hood** on the concept CAR. This is a relationship between rolesets where the more specific roles inherit all properties of the parent role except for Number Restriction. Differentiation can also occur locally on a concept.
- **particularization**: the roleset on an individual concept is related to a roleset on a parent generic concept. It is just like restriction except that it is on an individual concept. For example, the **Subparts** of **TUBE#5** are all CYLINDERs.
- **satisfaction**: this is the relationship between an IRole and its parent RoleSet defined by using a "**Sats**" link. For example, the **Engine** of TOYOTA#31 has the value TOYOTA-ENGINE66 and satisfies **Engine** of CAR.

Figure 4-3 provides examples of restriction, differentiation and satisfaction. The role **Subpart** on concept MOTOR-VEHICLE is modified to be a VEHICLE-PART. This

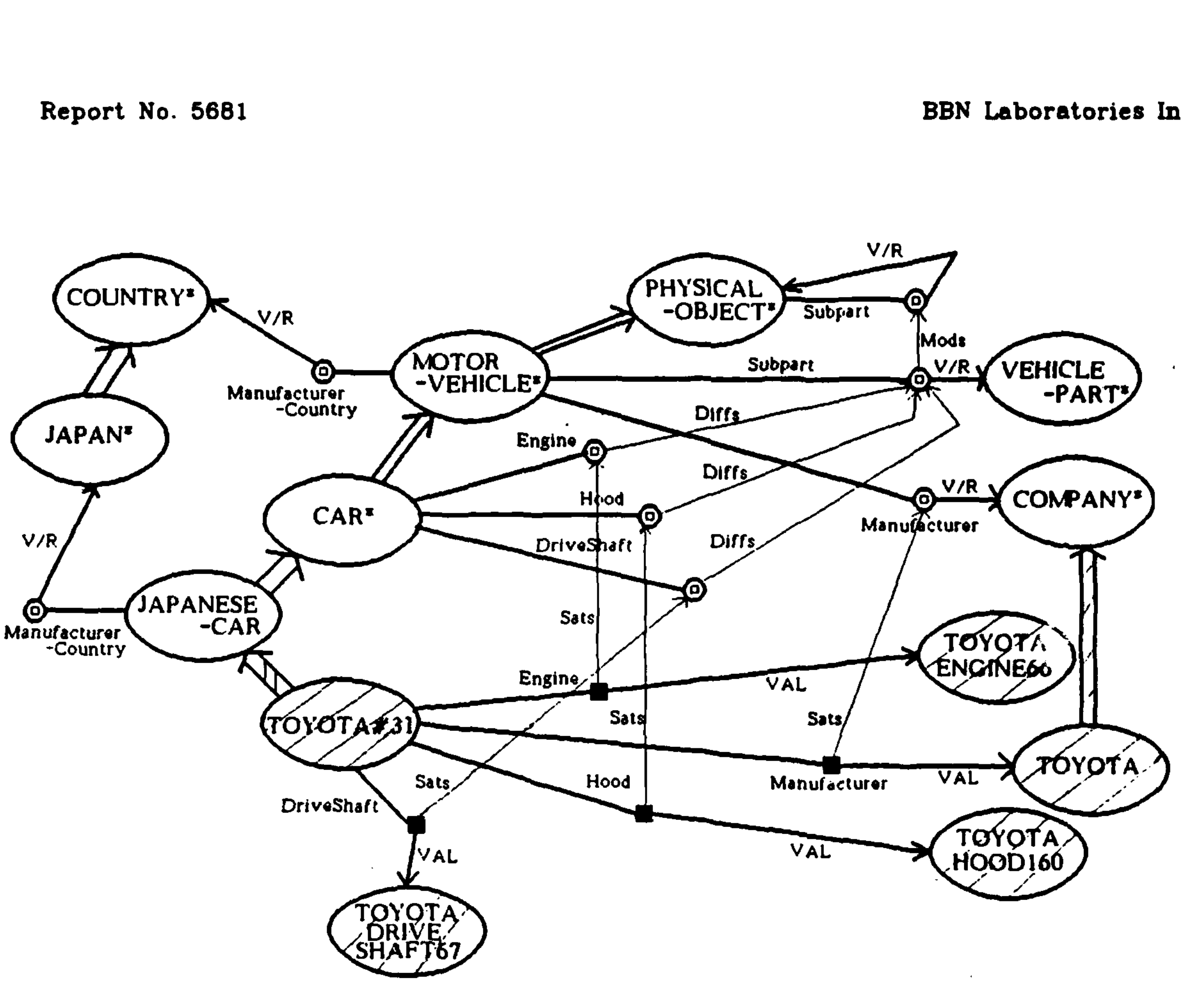


Figure 4-3: Example of restriction, differentiation, and satisfaction

further restricts the **Subpart** of a MOTOR-VEHICLE from being a PHYSICAL-OBJECT to being a VEHICLE-PART. The concept CAR provides an example of differentiation. Here the role Subpart is differentiated into three subroles - **Engine**, **Hood**, and **DriveShaft**. Finally, the individual concept TOYOTA#31 demonstrates role satisfaction. Each Irole of TOYOTA#31 is shown to satisfy a roleset on CAR.

4.1.2 Classification in KL-One

One of the strengths of KL-One is its ability to automatically maintain the taxonomy of concepts. This process, called Classification, determines the proper placement of each new concept when it is added to the taxonomy. The KL-One Classifier [88, 39, 66], written by Thomas Lipkis at USC/ISI, determines all appropriate subsumption relationships between a newly formed concept and all other concepts in a given taxonomy. The Classifier, where necessary, removes and installs appropriate

SuperC links. If the new concept turns out to be identical to a concept already present in the taxonomy, the new one is "merged" into the old one. A newly placed concept, thus, is guaranteed to be positioned below all concepts that definitionally subsume it and above all concepts that it subsumes. This strict enforcement of where new concepts are placed gives KL-One much of its power for representing and using knowledge compared to other knowledge representation systems. It also provides an important inference tool to systems using KL-One.

Consider the sample taxonomy shown in Figure 4-3. The taxonomy defines a car manufactured in Japan, JAPANESE-CAR, and it shows an individual car that is manufactured in Japan by Toyota, TOYOTA#31. It, however, is missing a generic concept to represent any car manufactured in Japan by Toyota. Such a concept, call it TOYOTA-CAR, would have a SuperC link to CAR, a role Manufacturer-Country whose V/R would be JAPAN, and a role Manufacturer whose V/R would be TOYOTA. When placed into the taxonomy in Figure 4-3, it would not show that it is a kind of JAPANESE-CAR and that TOYOTA#31 is a particular TOYOTA-CAR. Classification will discover this information; it will install a SuperC link between TOYOTA-CAR and JAPANESE-CAR; it will remove the SuperC link between TOYOTA#31 and JAPANESE-CAR; and it will install a SuperC link between TOYOTA#31 and TOYOTA-CAR.

The Classifier makes a distinction between primitive concepts (which are marked with a "*") and non-primitive concepts. Since primitive concepts are not fully-defined, it can not tell whether or not a new concept should be placed below it since it wouldn't know if information on the new concept would be inconsistent with the information missing on the primitive concept. The Classifier, hence, does not bother to check to see if a new concept can be placed below a primitive concept. It simply places the new concept as low as possible without putting it below the primitive concept.

4.1.3 Representing the water pump objects in the real world

The real world is a world that models the physical environment as it might be seen by a person or a vision system. The water pump objects in the physical world are three-dimensional and they are perceptible. A simulation of a person manipulating and identifying objects in that world requires representing basic perceptual information about those objects. I chose a representation strategy by considering the basic goal of my reference system (to identify objects in the world from a speaker's descriptions), the water pump assembly task itself, and the kind of input medium under which the representations could be formed (such as a vision system).[15] I felt the task required knowing the basic dimensions of an object (such as its size or volume), a more explicit description of the object that provides shape information, physical aspects of the object (such as color, transparency, weight and other physical features), and simple functional information. For this reason, a distributive (multi-view) approach to describing an object seemed appropriate. This allows each view to be simpler, making it easier to use the representation, at the expense of putting more strain on the mechanism that forms the representation of the object.[16] This approach was very conducive to the environment provided by KL-One.

4.1.3.1 Size and shape

The water pump objects are represented by two basic 3-D shapes: parallelepipeds (e.g., blocks, cubes or pyramids) and generalized cylinders (e.g., cones or cylinders) [1, 44, 45, 46, 2]. The parallelepiped is used for two different purposes. First, it is used to provide a sketchy representation of an object by forming the smallest block that fits around an object. This representation is of use when considering the object in terms of its *size* and *volume*. This makes it useful in deciding whether or not a particular object can fit in a space of a certain size or for comparing the size of two, possibly dissimilarly shaped, objects. Second, the parallelepiped is employed, along with the generalized cylinder, as a basic building block to use when representing an object. The combination of parallelepipeds and generalized cylinders in the representation provides a representation that more closely approximates the object's true shape. I chose the use of generalized cylinders and parallelepipeds since some vision systems [1, 44] have employed them in their recognition schemes, showing that it is not unreasonable for me to assume that somehow I could get a vision system to provide me with such a representation of a water pump object.

[15] The representation scheme described in this section is strongly influenced by the work of Gerald Agin [2] on representing 3-D objects.

[16] Which, in this case, turned out to be me since I represented the water pump objects by hand.

A generalized cylinder is described along a central axis by defining the cross-section at each point – the cross-section being defined by a function that keeps the shape the same but that can vary the size. For example, a function that varies the cross-section diameter of a circle linearly down to zero forms a cone while one that keeps the circle's diameter constant forms a cylinder.

Given such a set of building blocks, the algorithm for describing an object involves putting together the minimal set of parallelepipeds and generalized cylinders to conform to the shape of the object. The method for performing the segmentation involves: (1) trying to find a central, primary section of the object to use as a base for centering the other segments of the object around; (2) choosing the basic shape that best models this central section; (3) orienting that section in the "standard orientation" of a 3-D Cartesian coordinate system, with the section placed on the z-axis with its bottom sitting on the x-y plane and the z-axis running along its central axis; (4) scanning up the central axis (the z-axis) and choosing a basic shape to represent each segment;[17] and (5) trying, for each of the parts of the 3-D object that were segmented, to recursively apply this segmentation scheme to see if they can be segmented further.

For example, consider the part of the water pump called the *MAIN-TUBE* that is shown in Figure 4-4. It is a long cylindrical object that has four openings. The axis is selected to run through the tube from top to bottom. An examination of the tube from top to bottom, following along the central axis, yields five major segments of the tube. Each of these segments are cylinders as shown in Figure 4-5. The cylinders that represent the side openings (Outlet1 and Outlet2) intersect the long tube (Tube) and must be rotated and translated accordingly from their standard orientation.

Figure 4-6 shows a representation in KL-One of the tube shown in Figure 4-4.

[17] There are a few heuristics that can be used for deciding whether or not to form a new segment: (1) by definition of generalized cylinders and parallelepipeds, each segment should be defined as an object whose cross-section varies in a uniform manner along some axis through the object – so look for this uniform change (i.e., examine cross-sections) [1, 2], (2) look for discontinuities – points where the cross-section jumps from its uniform pattern – as places that segment a boundary between basic objects, and (3) ignore projections coming out of the object in a direction "much" different than that of the axis you are currently scanning along.

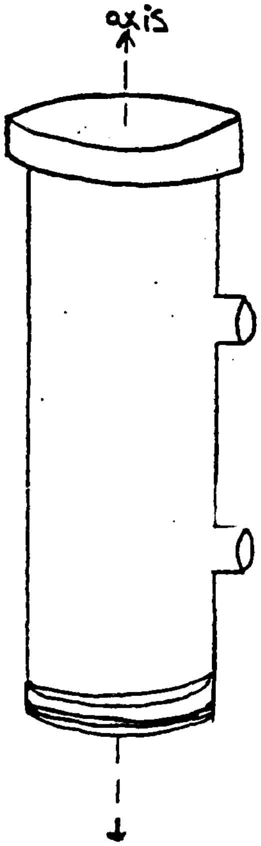


Figure 4–4: The main tube

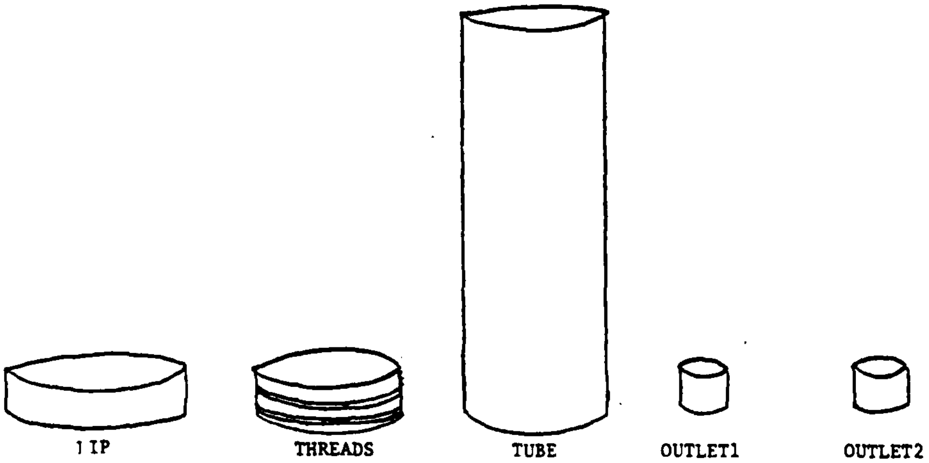


Figure 4–5: The Basic Pieces of the Tube

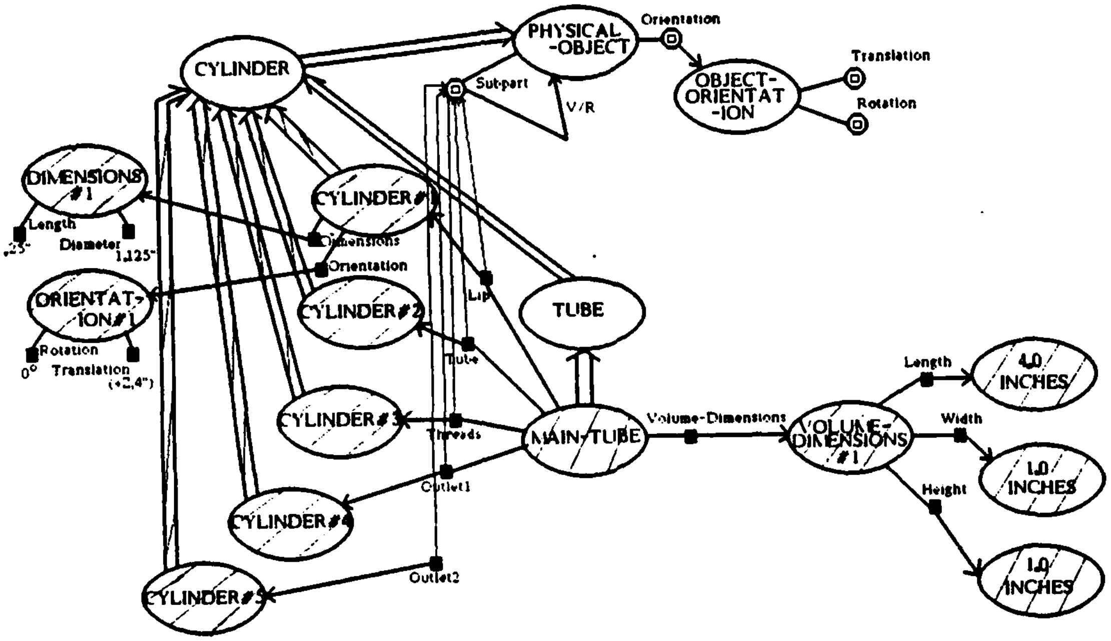


Figure 4-6: KL-One Representation of the Tube

The object is described in terms of its basic shapes. Here the basic shapes are five cylinders: LIP, TUBE, THREADS, OUTLET1 and OUTLET2. The positions of each basic shape of the object are given as a translation and rotation from their standard orientation. A translation is defined by giving the distance to move the object with respect to each axis. For example, {(+X,1"),(+Y,3"),(+Z,1")} defines movement of the object one inch in the positive x-direction, 3 inches in the positive y-direction and one inch in the positive z-direction. A rotation is denoted by giving the angle to move the object about an axis. For example, {(+X,30°),(-Z,60°)} rotates the object 30 degrees around the x-axis in the positive direction and 60 degrees around the z-axis in the negative direction.

4.1.3.2 Physical properties of an object

Some physical properties of an object that are useful in describing it include COLOR, TRANSPARENCY (whether or not you can see through the object), COMPOSITION, STRENGTH, WEIGHT, and its state of MATTER (gas, liquid or solid). These features are useful in providing a way for distinguishing one object from another possibly without examining the more complex size, shape and function descriptions.

The difficult part in defining physical properties is deciding what to include as their legitimate values. The problem is that it is impossible to predict what level of specification is sufficient, in general, for most objects. The level of specification required is based not only on the particular physical property itself but on the task being performed to the object. Below I present a classification of those physical features to be used in the water pump assembly task. It is not meant to be inclusive.

COLOR: [black;violet;purple;blue;green;
pink;red;colorless]

DIMENSIONS: [{numerical measure}]

ORIENTATION: [{numerical measure}]

THICKNESS: [{numerical measure}]

TRANSPARENCY: [clear;translucent;opaque]

COMPOSITION: [plastic;rubber;metal]

WEIGHT: [{numerical measure}]

STRENGTH: [hard;soft;flexible]

MATTER: [gas;liquid;solid]

4.1.3.3 Representing functional information

The kind of information needed to represent the functional aspects of an object can be quite broad depending on the use of the object and the actions that can be performed to it. In the water pump domain, I considered only a few simple functional properties such as **containment** (i.e., a *CONTAINER*), **flow** (i.e., a *TUBE*, *OUTLET*, or *SPOUT*), **capping** (i.e., a *CAP* or *VALVE*), and **attachment** (i.e., actions such as *PUSH-INTO*, *PUSH-ONTO*, *TWIST-INTO*, *TWIST-ONTO*, *SCREW-INTO*, and *SCREW-ONTO*). Under these definitions, a tube, for example, is defined as a cylinder that is also a functional object. I don't actually try to reason about the specific things one can do with a tube but treat it as a primitive in my system. Figure 4-7 provides a KL-One representation of the tube represented in Figure 4-4. It describes the tube using size and shape, physical properties and functional information.

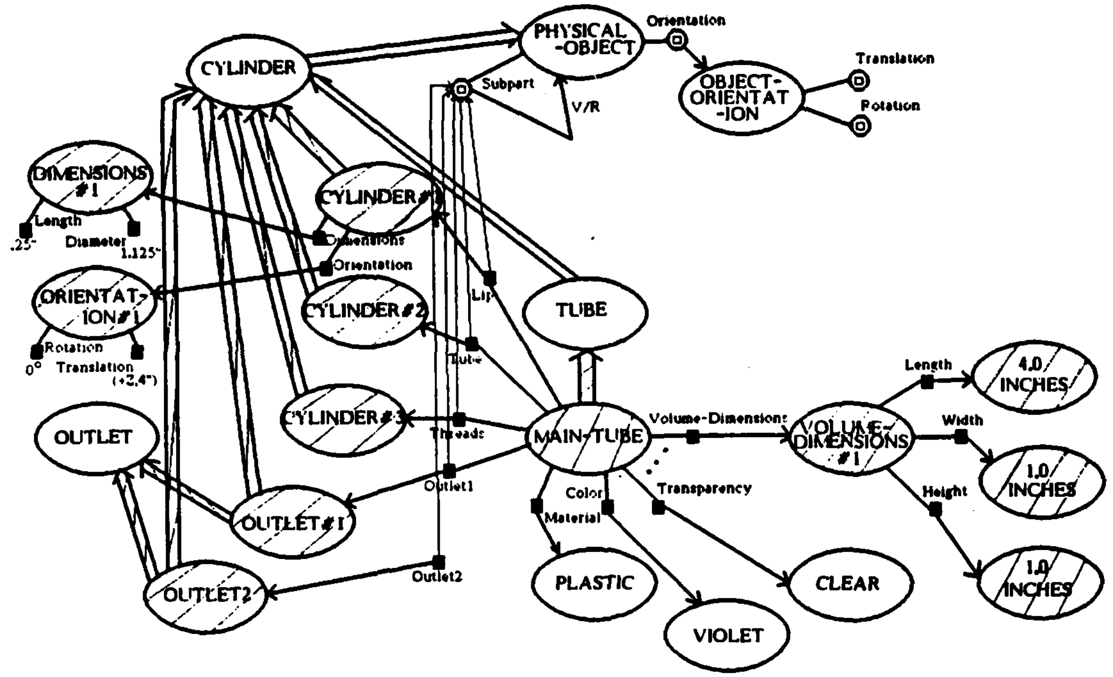


Figure 4-7: KL-One representation of the MAIN-TUBE

4.1.4 Representing the water pump objects in the linguistic world

The previous section described a way to represent the water pump objects as found in the real world. That representation, however, ignores how humans really talk about such objects. The linguistic world is meant to fill that gap by describing the parts and features of an object in linguistic terms. In many ways it simulates the result of human skill to extract information from our perceptual system and turn physical representations into words. It is more suggestive of a person's own perceptions and represents the words people use to describe an object.

This representation is very critical in the reference identification task because hearers are given a speaker's verbal description of an object and not handed his actual perceptual input. Hearers, thus, are provided with the speaker's interpretation and biases about his perceptual field. The linguistic world in reference, hence, represents more about the speaker's world than it does about the hearers. Reference identification can be defined as the task of determining whether an element in the linguistic world and an element in the *real world* co-describe the tangible object in the physical world. Since the tangible object isn't really available to my system, this task reduces to finding a description in the real world that is described by the linguistic world description.

A representation scheme for the linguistic world is basically a superset of the real world one because a person could describe an object almost exactly like he perceived it (i.e., with enough detail and precision, a speaker could describe his perceptual view of an object to the listener). It, thus, could describe a cylinder by a definite set of dimensions – its *length* and *diameter*. People, however, often describe the cylinder using less precise terms such as relative sizes like "big," "large," and "long," so such terms are part of the linguistic world. The real world is composed of 3-D shapes (e.g., generalized cylinders) while the linguistic world allows one to describe an object using analogical shapes (e.g., "the L-shaped tube"). Figure 4-8 shows a linguistic world representation of the tube described in Figure 4-4. This is in contrast to the real world description of the same tube shown in Figure 4-7. Notice how the role DIMENSIONS in the real world description is replaced by the role SIZE in the linguistic world description.

4.2 Parsing and semantic interpretation

The BBN system employs much of the methodology found in the TDUS system. It has a parser, a semantic interpreter, and a knowledge base. The parser is the RUS parser [6] which has evolved from the LUNAR parser. The RUS parser works in conjunction with the PSI-KLONE semantic interpreter [9, 7]. The knowledge base is represented in KL-One.

The RUS parser's primary improvement over the LUNAR parser is that it closely ties syntax and semantics together, allowing parsing to proceed in parallel with the semantic interpretation.[18] This differs from the LUNAR approach where the syntactic

[18] Actually the current implementation of the parser and semantic interpreter run as a cascade but they are designed to run in parallel. A related approach is used in the DIAMOND/DIAGRAM work at SRI [54, 62].

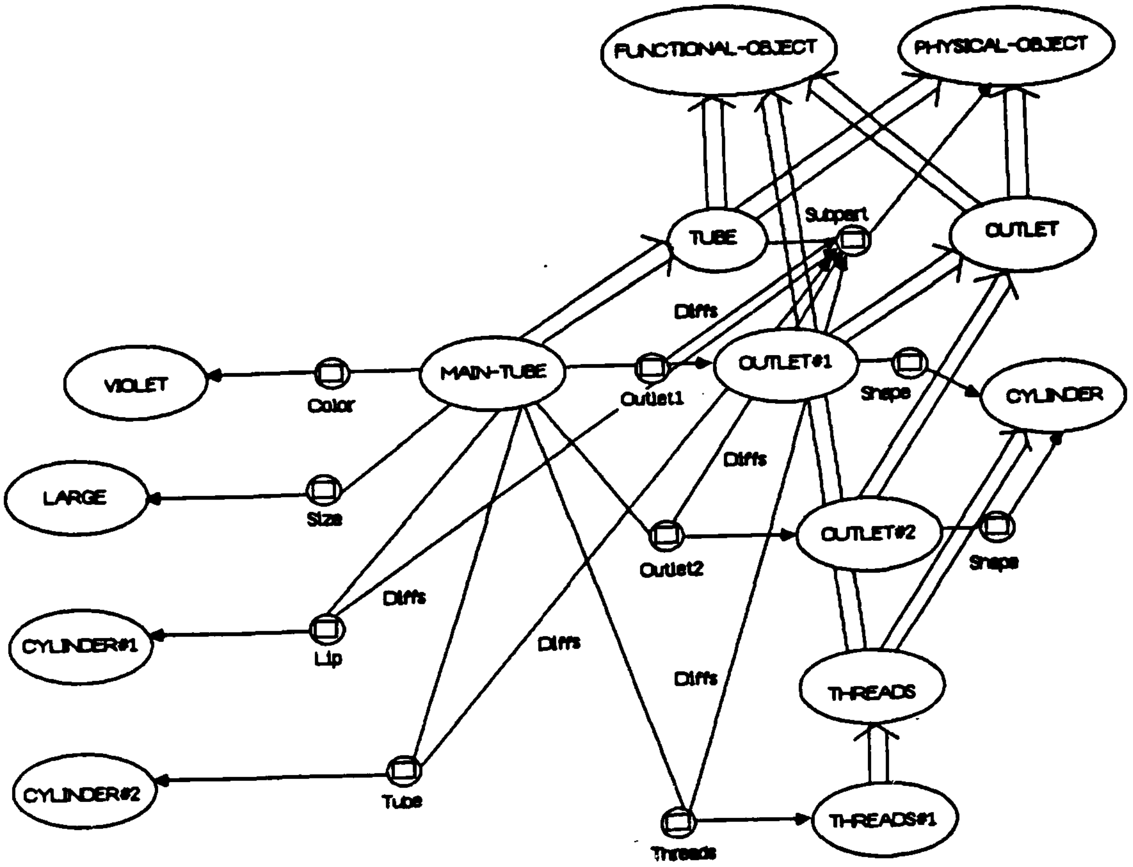


Figure 4-8: The MAIN-TUBE represented in the linguistic world

and semantic components were distinct entities in the system that worked one after the other. The advantage is that the parser combines the efficiency of a semantic grammar [13, 25] with the flexibility and extensibility of separate syntactic and semantic components. The parser can make use of semantic constraints, often avoiding trying unnecessary parses. This makes the parser much more efficient.

The parser and semantic interpreter avoid operating sequentially by communicating back and forth as parsing proceeds.[19] When the parser recognizes a constituent, it presents the interpreter with the constituent along with a proposal as

[19] While my referent identifier described later in this chapter operates after parsing and semantic interpretation are completed, there is nothing in its design to preclude it from operating in parallel, too. This makes its design more faithful to the actual data in the protocols.

to where it should be attached in the parse tree. The semantic interpreter examines this and decides whether or not to accept the parser's proposal. If the interpreter accepts the proposal, it returns a representation of the semantic knowledge of the phrase.

The mechanics of transmitting a constituent and a proposed place to attach it in the parse tree, from the parser to the semantic interpreter, is done by adding the transmission task to an ATN arc (see Figure 4-9 below). That way, should the transmission action fail because the semantic interpreter rejects the parser's proposal, the arc will also fail, causing the other arcs to be examined or backup to occur.

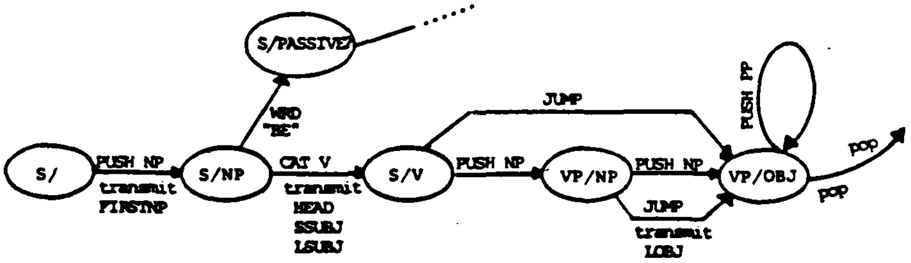


Figure 4-9: A simplified ATN for clauses

Consider the description "the large violet tube with two cylindrical outlets." The KL-One network that gets constructed by the parser to represent syntactic and semantic aspects of the description is shown in Figure 4-10. It is called a "syntaxonomy." The shaded concepts and arrows represent the actual instantiation of the parse of the description. The concept TUBE-NP#1 is the central concept representing that description. The concepts whose names are shown between two backslashes ("\...\") represent the word that corresponds to that name. The other (non-shaded) concepts are the part of the knowledge base that definer the kinds of utterances that RUS and PSI-KLONE know about. These include noun phrases (NP), adjectives (ADJ), prepositional phrases (PP), prepositions (PREP), determiners (DET) and so forth. It also describes how such terms are used. For example, it notes if the word is the "head" of a noun phrase (Head), whether it acts as an object (Obj), and so on.

RUS and PSI-KLONE place the description, with the aid of the KL-One Classifier [39], into the network. Once there, semantic interpretation rules can fire to perform the actual interpretation of the description.

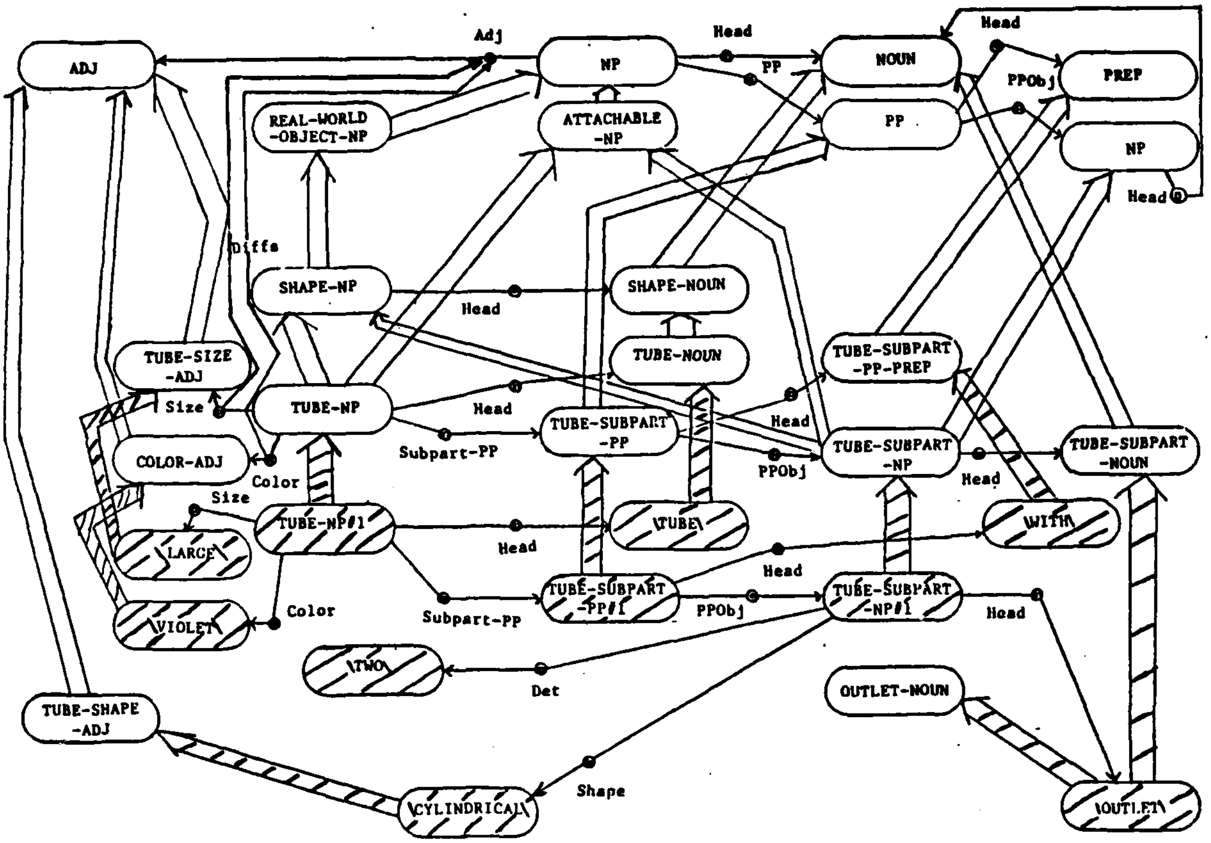


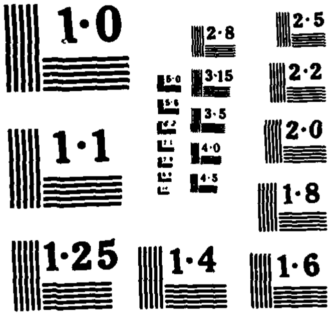
Figure 4-10: A sample syntaxonomy

Figure 4-11 contains the set of semantic interpretation rules associated with the example "syntaxonomy" network in Figure 4-10.[20] These rules are attached to the concepts and roles that represent RUS and PSI-KLONE's knowledge about descriptions. They can be inherited by concepts and roles that are lower in the network (such as those, like TUBE-NP#1, created to represent the current description). The rules are implemented as either a role (an INTERP role's value restriction is the interpretation

[20] These are actually represented as part of the network but were left out to reduce the clutter in the figure.

AD-A162 843   COMMUNICATION AND MISCOMMUNICATION(U) BOLT BERANEK AND   2/3
NEWMAN INC CAMBRIDGE MA   B A GOODMAN OCT 85 BBN-5681
N00014-85-C-0079

UNCLASSIFIED   F/G 9/2   NL


1·0
2·8
2·5
3·15
2·2
5·0
5·6
3·5
1·1
2·0
4·0
4·5
1·8
1·25
1·4
1·6

```
|C|SHAPE-NP:  INTERP is |C|SHAPE

|C|TUBE-NP:  INTERP is |C|TUBE
|R|SHAPE of |C|TUBE-NP:  (SHAPE INTERP)==>(INTERP SHAPE)
|R|COLOR of |C|TUBE-NP:  (COLOR INTERP)==>(INTERP COLOR)
|R|SIZE of |C|TUBE-NP:  (SIZE INTERP)==>(INTERP SIZE)
|R|SUBPART-PP of |C|TUBE-NP:  (SUBPART PPOBJ INTERP)==>(INTERP SUBPART)

|C|TUBE-SHAPE-ADJ:  INTERP is |C|CYLINDRICAL

|C|SHAPE-NOUN:  INTERP is |C|SHAPE

|C|\VIOLET\:  INTERP is |C|VIOLET
```

Figure 4-11: PSI-KLONE semantic interpretation rules

of the concept on which the role is present) or data attached to a role (the attached data provides a rule on how to interpret the role). The rules are read as follows. The left-hand side shows the entity that is being interpreted (either a concept or a role on a concept) and the right-hand side is the interpretation. It is here that the jump is made from words to concepts about physical objects. For example, the second rule in Figure 4-11 states that noun phrases that are parsed and placed under the concept TUBE-NP (i.e., **|C|TUBE-NP**) are interpreted as the concept TUBE, where TUBE is defined to be a kind of physical object (i.e., "the interpretation of TUBE-NP is TUBE"). The fourth rule states that "the interpretation of the color is the color of the interpretation." In simpler terms, the rule states that the role COLOR (i.e., **|R|COLOR**) on the concept TUBE-NP is interpreted to be the role COLOR on the interpretation of TUBE-NP, which is TUBE. Part of the interpretation of a role is the interpretation of the value restriction on the role. In this example, the value restriction of **|R|COLOR** of **|C|TUBE-NP#1** is **|C|\VIOLET**. The interpretation of **|C|\VIOLET** is **|C|VIOLET** (i.e., the word "VIOLET" is interpreted to be the physical color "VIOLET"). The sixth rule is more complex. Here the semantic interpretation looks deeper than the value restriction **|C|TUBE-SUBPART-PP** on role **|R|SUBPART-PP**. It jumps directly to the value restriction on role **|R|PPOBJ** of **|C|TUBE-SUBPART-PP**, i.e., **|C|TUBE-SUBPART-NP**. This avoids unnecessarily embedding the interpretation. The rule says that "the interpretation of the PPOBJ of SUBPART is the SUBPART of the interpretation." In simpler terms, the rule states that the role SUBPART-PP on the concept TUBE-NP is interpreted to be the role SUBPART on the concept that is the interpretation of the role PPOBJ on the concept TUBE-SUBPART-PP. The complete interpretation of the phrase "the large violet tube with two cylindrical outlets" yields the network structure shown in Figure 4-12. Notice that it is not here, however, that the system finds the actual object in the world that corresponds to the description (instead the system simply built a "template" that can be used to search for the real object). That is delayed until the referent identification stage.

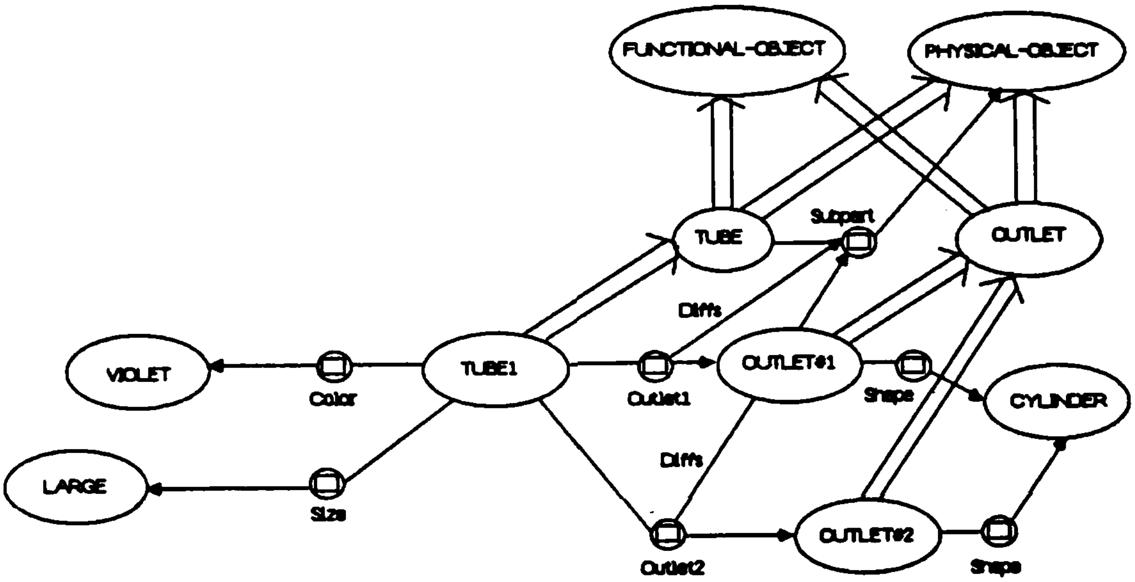


Figure 4-12: A sample interpretation

4.3 Reference identification

The last section discussed parsing and semantic interpretation. It described how a speaker's description of an object is turned into a KL-One structure that represents it. That description is a linguistic world element and <u>not</u> a real world one because it conforms to the speaker's <u>interpretation</u> of his real world view. The description is partially specified because a speaker tries to convey just enough salient information to allow a listener to find the referent in the listener's real world. If it fit exactly the speaker's real world view, then the reference identification task would be much simpler because a speaker's own biases and perceptual abilities wouldn't be reflected in the description. The listener places the speaker's description in his own linguistic world knowledge base and then uses that description as a template to search the real world. My system attempts to simulate this search. The simulation behaves differently depending on the complexity of the speaker's description. There are two cases, both which use the speaker's partial description as a template: (1) the speaker's description contains no complex components that require subjective evaluation on the part of the listener (i.e., it doesn't use complex features such as relative dimensions) or (2) the speaker's description contains complex components that require subjective evaluation. In the first case, the reference identification system need only search once to find a referent. The second case, however, requires "two" searches. The first search ignores the parts of the description that require evaluation and attempts to determine if anything in the world fits the description. The second part of the search then tries to use the more complex components of the speaker's description to determine exactly which element is the referent (assuming the first search didn't already determine that nothing could fit).

The basic search mechanism uses the KL-One Classifier [39] to search the real world knowledge base taxonomy. The Classifier's purpose is to discover all appropriate subsumption relationships between any newly formed descriptions and all other descriptions in a given taxonomy [39]. With respect to reference, this means that all possible referents of the current interpreted description will be subsumed by it after it has been classified into the knowledge base taxonomy. If more than one referent candidate is below the classified description, then, unless a quantifier in the description specified more than one element, the speaker's description is ambiguous. If exactly one description is below it, then the intended referent is assumed to have been found. Finally, if no referent is found below the classified description, then something may be wrong with the description.

For example, consider the description "the violet tube." The linguistic world representation of that description, as created by the parser and semantic interpreter, can be seen on the left side of Figure 4-13. The search for the referent is achieved by making a copy of the linguistic world description - call it *PROBE* - and then classifying it into the real world knowledge base (which is shown on the right side of Figure 4-13). The Classifier compares *PROBE* to *TUBE1* and *MAIN-TUBE*. It can't place *TUBE1* below *PROBE* because the V/R of role COLOR on *TUBE1* is *"BLUE"* while it is *"VIOLET"* on *PROBE*. It can, however, place *MAIN-TUBE* below *PROBE* since they both have a V/R of *"VIOLET"* on their respective COLOR roles. The result of the

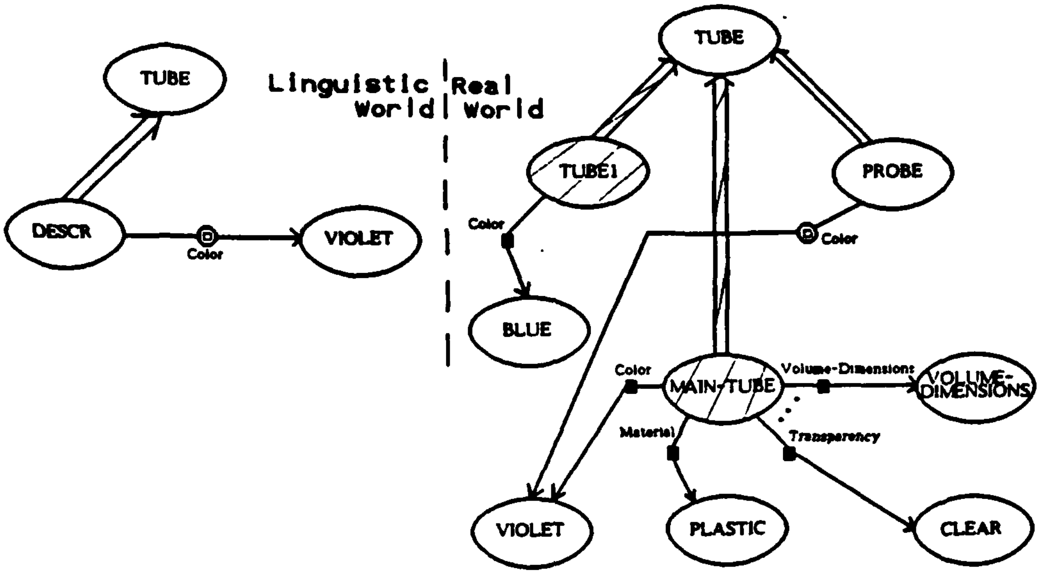


Figure 4-13: A simple referent search

classification can be seen in Figure 4-14. Since *PROBE* subsumes *MAIN-TUBE*, the referent of *PROBE* is *MAIN-TUBE*.

The basic classification process is a little more complex than I just described. There can be features in a speaker's description that require further processing before they can be compared against descriptions of elements in the real world. These include things like superlatives (e.g., "largest," "longest," or "the most"), comparatives (e.g., "larger," "longer," or "more"), and relative dimensions (e.g., "large," "long," or "thick"). In those cases, the system manually pushes the classified template down further by using a special routine that determines if the condition holds. Currently these are menu-driven routines that simulate the checking by asking the user whether or not a particular condition holds. For example, a routine may ask if the object is "large" compared to a group of other objects. The user's response will determine whether or not the classified description is placed any lower in the taxonomy. Heuristic routines could be implemented that try to determine on their own whether the condition holds. Appendix D provides a more detailed description of the routines for handling superlatives, comparatives and relative dimensions.

For example, consider the description "the large violet tube." The left side of Figure 4-15 gives the linguistic world representation of the description. Notice that

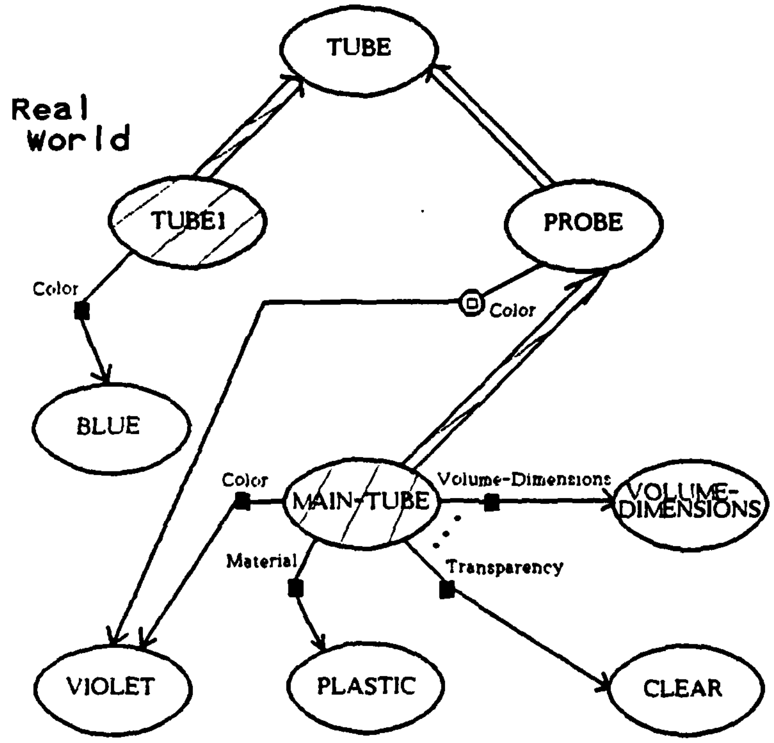


Figure 4-14: The classified PROBE

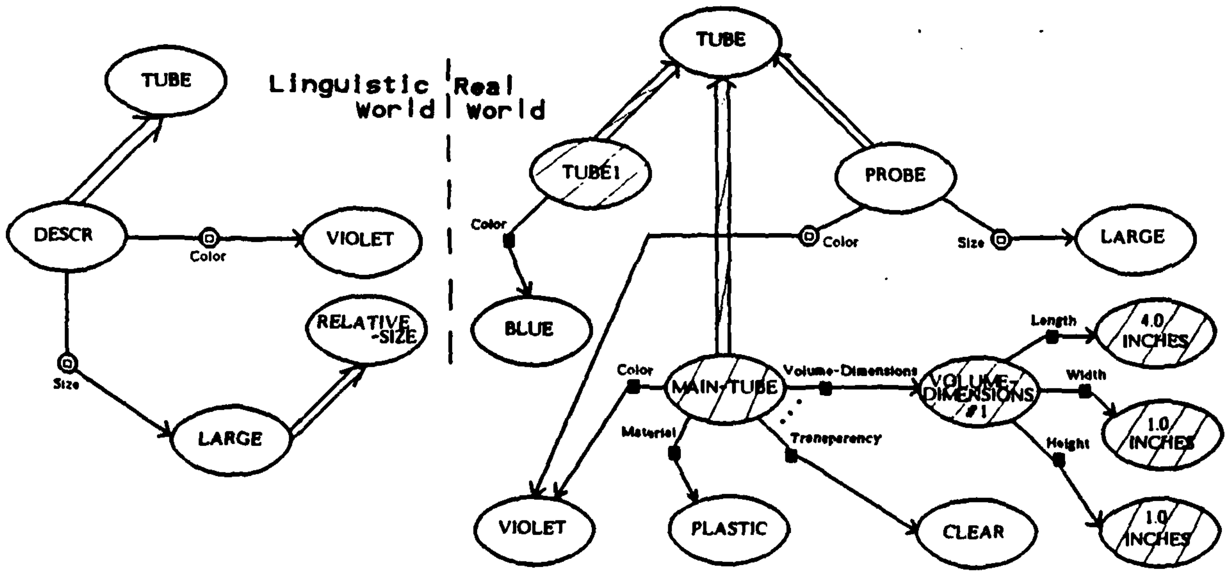


Figure 4-15: A more complex referent search

this description is more complex than the one in the previous example because it describes the size of the tube using a relative size value, "large." Such relative values introduce a difficulty to the system because they can't be compared by the Classifier directly to size values in the real world descriptions. This means that the Classifier at the best will only be able to place the description part way down into the taxonomy. For this example, the *PROBE* description cannot be moved any deeper into the taxonomy by the Classifier. A routine for handling relative size is invoked and it compares the V/R "*LARGE*" on *PROBE* to the V/R of the role VOLUME-DIMENSIONS on *MAIN-TUBE*. The comparison determines that *MAIN-TUBE* is "large" and, since the COLOR role on *PROBE* and *MAIN-TUBE* both have V/R "*VIOLET,*" it removes the SuperC link between *MAIN-TUBE* and *TUBE* and places one between *MAIN-TUBE* and *PROBE*. This means that *MAIN-TUBE* is the referent of the description. The resulting configuration of the real world taxonomy is shown in Figure 4-16.

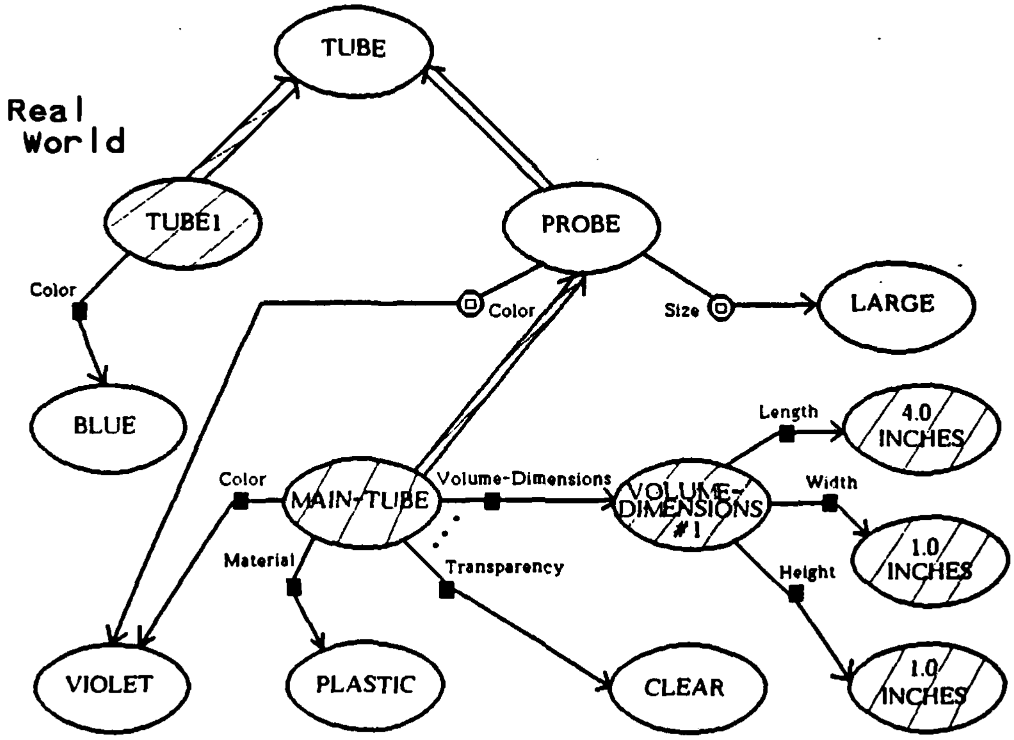


Figure 4-16: The classified complex PROBE

4.4 Focus

Focus considerations can be added to the reference identification mechanism to further constrain the search space for a referent. I follow the Grosz [30, 32] approach to focus illustrated in the last chapter. There are, however, some important differences. Grosz assumes that, in most cases, a speaker and listener share a common focus and that they don't have distinct models of each other's focus. I make no such assumptions since I am interested in the misunderstandings that result when the speaker and listener don't share a common focus. Grosz also assumes that the hearer has as much knowledge of the element in focus as the speaker. I don't make this assumption since it is one of the reasons that conversants miscommunicate.[21]

There are also implementation differences between the Grosz focus mechanism and my own. I needed to expand the representation of focus to help detect miscommunication due to focus problems. The real world knowledge space is basically equivalent to a KVISTA, describing the current physical world in front of the listener, but the linguistic world knowledge space differs from a QVISTA. The linguistic world is used to track information from previous utterances such as their semantic interpretation so that it is simpler to detect focus shifts, reference errors related to focus problems, and to handle anaphoric descriptions. It also makes it possible to access only those properties of the focused element that have been mentioned so far. This simplifies detection of focus shifts and some misreference problems. I also generate two, distinct sets of focus partitions – one set for the real world and one for the linguistic world. This makes it easier both to detect focus problems and to isolate the source of the problem.

There are also some slight differences in terminology between the focus work of Grosz and my own. I use "context" instead of "focus space" to describe all the linguistic elements that refer to elements that are currently in explicit focus.[22] I then divide up the context into focus partitions that each hold a set of linguistic elements that uniquely refer to one real world element that is explicitly in focus.

[21] Sidner [69] also points out this problem.

[22] See Reichman [57] for a similar use of "context."

4.4.1 Extending the representation to handle focus

Focus is used in my system to provide two related partitions. One group of partitions divides up the KL-One representation of the linguistic world while the other group separates parts of the KL-One real world representation. The actual representation of the water pump objects in the real world and the linguistic world were described, respectively, in sections 4.1.3 and 4.1.4. This section attempts to augment those representation schemes to handle focus.

The linguistic world partition has two purposes. One is to provide a way to group a set of utterances and the other is to supply a differential access path into the real world knowledge base. The partitioning is achieved by creating two levels of KL-One concepts that are used to organize the linguistic world. The first level is defined by the CONTEXT concepts, such as shown in Figure 4-17, which are created at the beginning of the dialogue or after a focus shift to represent the fact that a new global focus has been created.

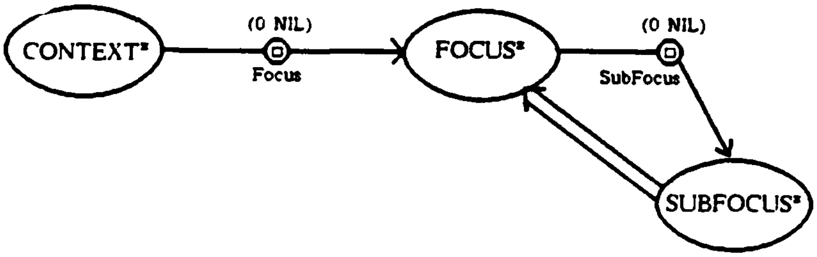


Figure 4-17: The CONTEXT and FOCUS concepts in the linguistic world

The CONTEXT concept's role, Focus, defines the second level of partitioning. That level is composed of a group of FOCUS concepts. Each of them corresponds to a set of linguistic descriptions that all refer to the same real world object. Since a real world object can have subparts, the FOCUS concepts in the linguistic world can be similarly divided into a set of SUBFOCUS concepts.

Figure 4-18 shows an example of a description of a tube, *TUBE1*, represented in the linguistic world and Figure 4-19 shows a possible correspondent for it,

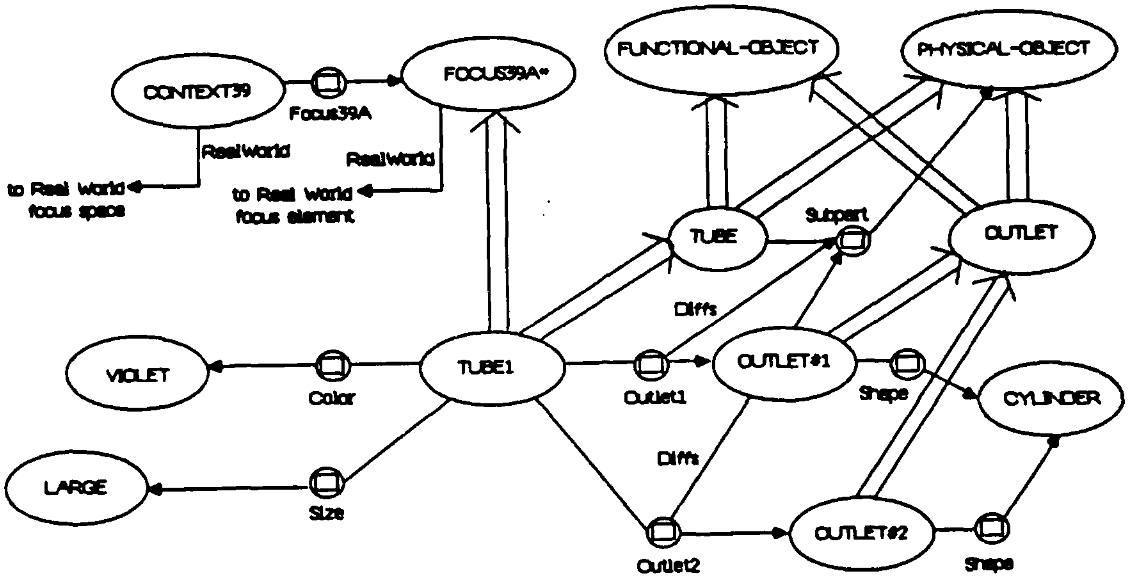


Figure 4-18: A tube represented in the linguistic world

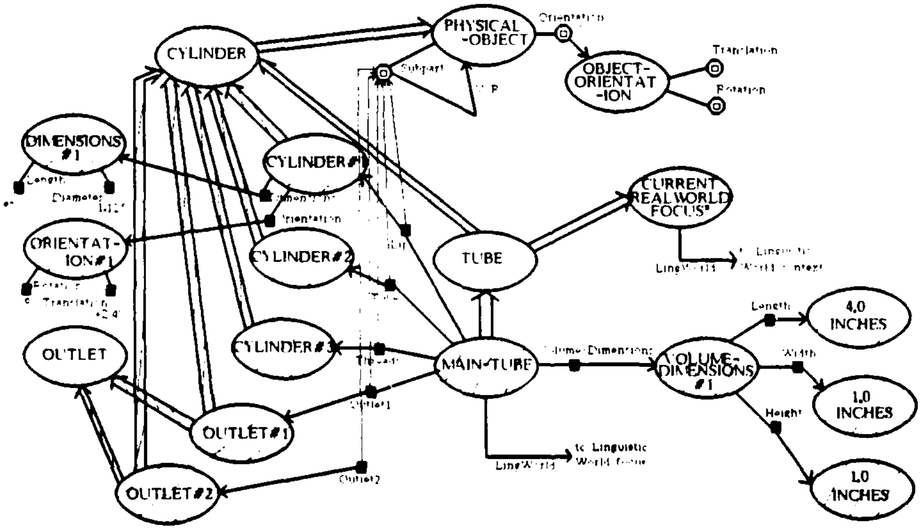


Figure 4-19: A tube represented in the real world

MAIN-TUBE, in the real world. Consider *TUBE1* in Figᵘʳe 4-18. The SuperC link between *TUBE1* and *FOCUS39A* states that *TUBE1* is in focus *FOCUS39A*. *FOCUS39A* is found, by following the inverse of the V/R that points to it, to be assigned to context *CONTEXT39*. The attached pointer *RealWorld* on *FOCUS39A* points to the real world element that is the referent of all descriptions in *FOCUS39A* (i.e., there can be other descriptions, say *TUBE2*, that are in the current focus - *FOCUS39A* - and that refer to the same real world element as *TUBE1*). The attached pointer *RealWorld* on *CONTEXT39* points to a focus space in the real world that corresponds to the linguistic world context.

Now consider the real world represented in Figure 4-19. It defines a partitioning analogous to the one shown for the linguistic world in Figure 4-18. Focus *CURRENTREALWORLDFOCUS* defines a focus space that contains the current objects in the real world that are explicitly in focus. An element is defined to be in a particular focus space if a SuperC link runs from the concept representing the element to the concept representing the focus space. In the figure, *MAIN-TUBE* is shown to be in focus *CURRENTREALWORLDFOCUS*. A pointer, *LingWorld*, is attached to focus *CURRENTREALWORLDFOCUS* and points to the correspondent context in the linguistic world. *MAIN-TUBE* also has an attached pointer, *LingWorld*, that points to the focus element in the linguistic world that groups together all the descriptions that refer to *MAIN-TUBE*.

The semantic interpretation of a new input is placed into the linguistic world partition that is currently in focus. This is achieved by adding a SuperC cable between the KL-One representation of the user's input and the KL-One representation of the current focus. The current focus itself is part of the current context partition. The context partition has a pointer to the correspondent real world focus space that describes the currently relevant real world elements. Another pointer, that is on the current linguistic world focus concept, points to the particular element that is the current focus of attention.[23] The newly installed input automatically inherits the pointer to the real world element that is currently in focus. Unless there is some major discrepancy between the input and the real world focus, the referent of the input is assumed to be that real world element. Any discrepancies hint that a possible shift in focus has occurred or that th^ speaker has made a mistake. Another

[23] At the beginning of the dialogue, all elements in the real world are considered relevant but no one element is the focus of attention.

hint at a focus shift occurs if the current input contains information that was not previously mentioned. For example, if all previous inputs never mentioned the color of the object and now it is referred to as "the red thing," then the speaker may be hinting that focus should shift to something else [30]. Such shifts are usually to a subpart of the object currently in focus, to another object, or to a subassembly that some previous action has built.

Shifts in focus in a dialogue result in the restructuring of the partitions of the real world and the linguistic world. A shift in focus, from the point of view of the real world, results in a set of objects becoming the center of attention. These relevant objects are partitioned into a *focus space*. A corresponding shift also occurs to partitions of the linguistic world. The linguistic world is partitioned in accordance with the real world partitions to make reference resolution more efficient and so that anaphoric references can be resolved. It makes referent identification simpler because it constrains the search space, allowing the most relevant objects to be checked first. It allows for the resolution of anaphoric definite noun phrases because the linguistic world contains a conglomeration of previous references to an object. If the current input fits in line with the previous ones (i.e., there are no discrepancies), then the anaphoric expression is assumed to refer to the same real world object as the previous ones.

A description of the actual focus mechanism can be found in Appendix C. The mechanism described there is a simulation of the focus machines designed by Grosz [30] and Sidner [69]. The next section will treat in detail what happens when the reference identification system receives a new input from the speaker.

4.5 Reference identification with focus

The last section discussed the focus mechanism used by the referent identifier. It described how the referent identification process is sh^rtened when the speaker's description refers to an object already in focus (which is what can occur with an anaphoric definite noun phrase). In those cases, no search of the real world focus space is necessary since the expected referent is already known. There are times, however, when this luxury does not exist. One such case is that of initial reference.

Initial reference is when an object is referred to for the first time (normally via an indefinite noun phrase but, especially for spoken language, sometimes with definite noun phrases) at the beginning of a conversation or when a shift is made to a new focus space.

For initial references, the system places the current interpreted description, from the parser and semantic interpreter, into the new linguistic world focus space. Since there are no other elements in this space, this is achieved by placing the description below the concept that represents the new linguistic focus space. The placement allows for the resolution of any future anaphoric references to the same element. Because this is the first element in the focus space, the reference system has no pointer to a particular object in the real world portion of the knowledge base that is known to correspond to the description's referent. It might, however, have a pointer to a correspondent focus space in the real world that contains the currently relevant objects (i.e., ones that are possible candidates for the referent). A copy of the interpreted description is generated to use as a template to probe against elements in the real world focus space.

Another exception to the standard reference process occurs with references to objects that are implicitly in focus. As I described earlier, the TDUS system examines those objects implicitly in focus as well as those explicitly in focus when looking for a referent. My reference mechanism, before giving up because no referent was found during classification, also checks elements in implicit focus. It examines the subparts of objects in the current real world focus space to see if they match the template generated from the speaker's description. If none do, then it assumes some sort of miscommunication has occurred.

4.6 An example

This section provides a detailed example of the reference mechanism in action. It follows the same example that I used throughout the previous chapter. I assume that the speaker's description to analyze is "the large violet tube with two cylindrical outlets." I will describe the identification of the referent of the description under two different conditions. The first one assumes that the description is uttered just after a focus shift to a new focus space has occurred (i.e., an initial reference). The second case assumes that no focus shift has occurred and that the description is uttered and followed by another description that is intended anaphorically.

Figure 4-12 shows the semantic interpretation of the initial description (*TUBE1*). Figure 4-18 illustrates how the description is represented immediately after a focus shift to a new focus space has occurred and the description is installed into the linguistic world taxonomy. Concept *CONTEXT39* represents the linguistic focus space. The concept *FOCUS39A* was created to represent the current focus of attention. It is the value restriction of role *Focus39A* on concept *CONTEXT39*. *TUBE1* is placed under concept *FOCUS39A* by attaching a SuperC link between *TUBE1* and *FOCUS39A*. There is no pointer from *FOCUS39A* to an entity in the real world because this is an initial reference. A pointer exists on *CONTEXT39*, though, that points to the corresponding real world focus space, *CURRENTREALWORLDFOCUS* (see Figure 4-19). A referent for the description can be found by creating a copy of *TUBE1* (shown in Figure 4-12) and placing it in *CURRENTREALWORLDFOCUS*. I will call this concept, *PROBE*. *PROBE* can now be classified as shown in Figure 4-20.

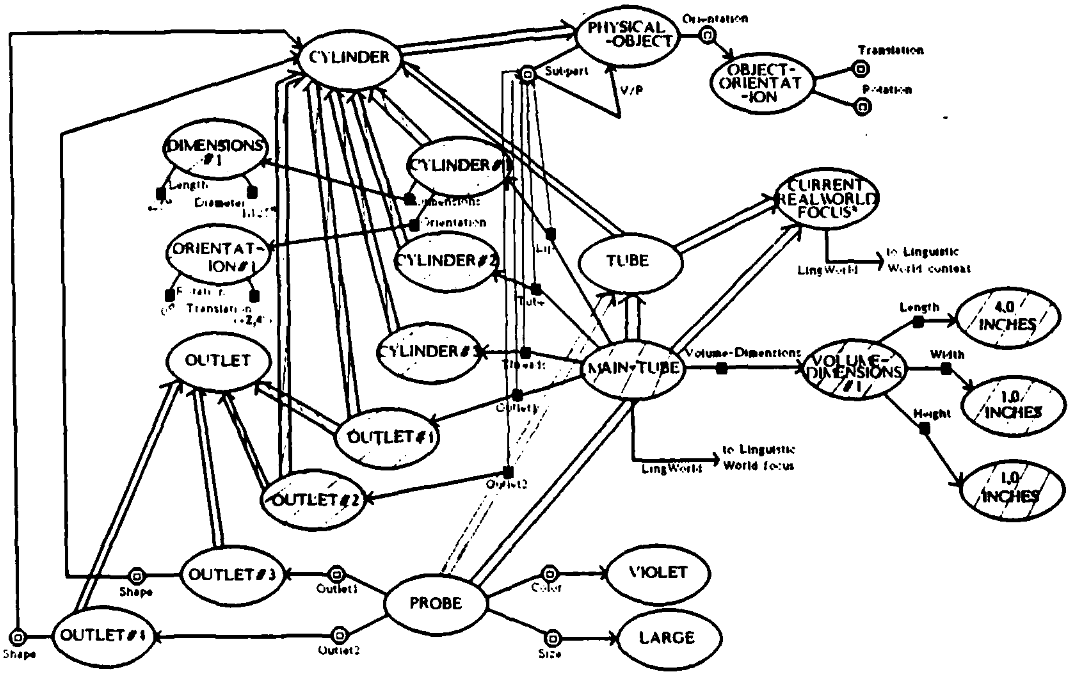


Figure 4-20: Probing for a referent

The Classifier discovers that the *COLOR* roles on *MAIN-TUBE* and *PROBE* are both *VIOLET* and that the subparts, represented as roles *Outlet1* and *Outlet2*, on *PROBE*, have correspondent subparts, roles *Outlet1* and *Outlet2*, on *MAIN-TUBE*. The subparts correspond not because of their equivalent role names, which are ignored by the Classifier, but because their V/Rs are both defined as *OUTLETs* and *CYLINDERs*. The Classifier, unfortunately, is unable to move *PROBE* any lower in the real world taxonomy because role *SIZE* on *PROBE* does not correspond to any of the roles on *MAIN-TUBE*. This problem occurs because *PROBE* describes the size of something by relative size while *MAIN-TUBE* uses numerical dimensions under the role *VOLUME-DIMENSIONS*. One of the special routines I mentioned in the last section and that is described in Appendix D can be used to resolve this difference. A menu is popped up that asks whether or not *MAIN-TUBE* is large (in particular, large with respect to the other objects in focus). If the user says that *MAIN-TUBE* is large, than *PROBE* can be moved lower in the taxonomy and a SuperC cable can be installed between *MAIN-TUBE* and *PROBE* (i.e., *PROBE* "subsumes" *MAIN-TUBE*). This means that *MAIN-TUBE* is the referent of my original description. A set of pointers are installed between the linguistic and real world elements to save this discovery. A pointer (*RealWorld*) between *FOCUS39A* and *MAIN-TUBE* is attached as data to *FOCUS39A*. A corresponding pointer (*LingWorld*) between *MAIN-TUBE* and *FOCUS39A* is attached to *MAIN-TUBE*. These pointers are used to make it easier to find referents for anaphoric descriptions.

Now I will consider the second case that occurs when an anaphoric description is used by the speaker. Assume that the current set of linguistic world and real world focus spaces are set up from the first example. If the speaker now utters the description "the violet tube," then the following occurs. The concept representing the new input (call it *TUBE2*) is placed under *FOCUS39A* since focus has not shifted. *TUBE2* is classified and this results in a SuperC link being placed between *TUBE1* and *TUBE2* and a SuperC link being removed from *TUBE1* to *FOCUS39A*. This means that *TUBE2* is not in disagreement with the previous description *TUBE1* (if it had, then a focus shift would have been implied). Finding a referent in this case, then, becomes trivial. *TUBE2* inherents from *FOCUS39A* the pointer to the object in the real world that is currently in focus – the *MAIN-TUBE*. This is the referent of description *TUBE2*.

4.7 Summary

This chapter laid down the foundation for the system that my reference identification mechanism is built around. It described the knowledge representation scheme shared by all components of the system and showed how its expressiveness allows for the representation of knowledge about syntax, semantics, physical objects, and discourse. I then described the basic reference identification and focus component of the system and demonstrated how a referent is found. The next chapter tries to address problems that can occur during reference identification and highlight the sources of knowledge that people use to get around such problems.

5. REPAIRING REFERENCE FAILURES

This chapter describes the language and physical knowledge that people use to perform reference identification and recovery from reference failure. The classification of knowledge sources and the observations on how to recover from reference failures were motivated from the analysis of the excerpts in Chapter 2. Those observations have been formalized as a set of relaxation rules that are used to determine when to delete or modify portions of a speaker's description. The last part of the chapter presents those relaxation rules.

5.1 Knowledge for repairing descriptions

When things go wrong during a conversation, people have many sources of knowledge that they bring to bear to get around the problem (e.g., see [60]). Much of the time the repairs are so natural that we aren't conscious that they have taken place. At other times, we must make an effort to correct what we have heard, or determine that we need clarification from the speaker. Either repair process involves the use of knowledge about conversation, social conventions and the world around us.

In this work, I chose to consider the repair of descriptions rather than complete utterances. The most relevant knowledge for repairing descriptions is the conversation itself and the real world described therein. This knowledge can be broken down into numerous forms. Linguistic knowledge is the knowledge that expresses the use of the structure and meaning of a description. Perceptual knowledge is composed of information about a person's abilities to distinguish feature values, his preferences in features and feature values (i.e., what features are most important to him in this domain), and his extraction of information from the internal representation of his perception of an object. Discourse knowledge is concerned with how a person interprets the flow of conversation and its effects on highlighting relevant parts of the world. Hierarchical knowledge is concerned with the use of knowledge about generality and specificity of descriptions to decide if a description is either too vague or overly specific. Trial and error knowledge is information gained when a listener attempts a requested action on requested objects and then compares the result of the action with his expectations. Other knowledge sources, such as pragmatic knowledge [18, 3, 55, 5], and domain knowledge [30] will not be covered here. Pragmatic knowledge is missing because I restricted my work to noun phrases instead of complete utterances. It would be more important when trying to work with complete utterances. Domain knowledge isn't covered because it is treated well elsewhere (e.g., see Grosz [30]).

These knowledge sources can be used to guide the repair of the speaker's description when no referent is found. They are part of a "relaxation" process. Relaxation would typically mean in the reference identification paradigm that the system drops features in the speaker's description one at a time until a referent is found or none are left. I have something different in mind. First, relaxation means more than simply dropping a feature value. It also means replacing the feature value with another one that the knowledge sources consider as reasonable. Second, I want an order to be chosen to drop the features. The interesting part is that this ordering comes from a negotiation among the knowledge sources. The actual negotiation, which is a control problem, is discussed in the next chapter.

5.1.1 Linguistic knowledge in reference

Speakers can utilize many different kinds of linguistic structures to describe objects in the extensional world. This section outlines some of these structures and their meanings and shows how they can be used to guide repairs in the description.

A description of an object in the extensional world usually includes enough information about physical features of the object so that listeners can use their perceptual abilities to identify the object.[24] Those physical features are normally specified as modifiers of nouns and pronouns. The typical modifiers are adjectives, relative clauses and prepositional phrases. They are often interchangeable; that is, one could specify a feature using any of the modifier forms. One modifier form, however, may be better suited for expressing some particular feature than another.

[24] Here I assume that either the speaker and hearer have a shared perceptual context or the speaker has an extensive model of the hearer's perceptual context.

Relative clauses are well suited for expressing complicated information since they are separate from the main part of the noun phrase and can be arbitrarily complex themselves. They can restrict the word or phrase they modify. They function in the following ways in extensional reference:

- Complex relationships such as spatial relations (e.g., "the blue cap that is on the main tube"), and function information (e.g., "the thing with the wire that acts like a plunger").

- Assertions of "extra" (usually restrictive) information, information possibly outside the domain knowledge and not useful for finding the referent at this time (e.g., "an L-shaped tube of clear plastic that is defined as a spout").

- Material useful for confirming that the proper referent was found (e.g., "the long blue tube that has two outlets on the side").

- A respecification of the initial description in more detail. For example, in the case of the descriptions "the thing that is flared at the top" and "the main tube which is the biggest tube," the relative clauses are needed because the initial descriptions are too general to distinguish any one object.

Prepositional phrases are better fitted for simpler pieces of information. They are often part of expressions of predicative relationships.

- A comparative or superlative relation (e.g., "the smallest of the red pieces").

- A subpart specification - used to access the subpart of the object under consideration (e.g., "the top end of the little elbow joint," "that water chamber with the blue bottom and the globe top").

- Most perceptual features (e.g., "with a clear tint," "with a red color").

Just like relative clauses, prepositional phrases can also provide confirmation information.

Adjectives are used to express almost any perceptual feature - though complex relations can be awkward. Usually they modify the noun phrase directly, but sometimes they are expressed as a predicate complement. In those situations, the complement describes the subject of the linking verb (e.g., "the tube is large"). As with some of the relative clauses above, predicate complements have an assertional nature to them because they are normally used to state something about the subject of a sentence.

Sometimes the head noun carries feature information. For example, one can use "the bell" to refer to a bell-shaped object (though it does not necessarily have the function of a bell), or can say "the cube" instead of saying "the block" to refer to an object.

It is implicitly clear that the structure of a noun phrase can affect its meaning in many ways (such as the ones mentioned above under relative clauses). Since there is no one-to-one mapping between a noun phrase's structure and its meaning, it is the hearer's job to determine how the structural information is being used.

5.1.2 Relaxing a description using linguistic knowledge

The relaxation process attempts to weaken or remove features in a description in the order: adjectives, then prepositional phrases and finally relative clauses and predicate complements. This order was chosen by examining the water pump protocols and noting where and when the linguistic forms come into play during reference resolution (i.e., I saw that people would often commence their search for a referent immediately, using each piece of the description as it is heard). Adjectives and prepositional phrases play a more central role during referent identification, because they are heard first, while relative clauses usually play a secondary role, because they normally come at the end of a description, often after a pause. However, relative clauses and predicate complements exhibit an assertional nature that, while reducing their usefulness for resolving the current reference, provides useful information that can be expressed in subsequent (anaphoric) references. For example, a speaker can describe the *MAIN-TUBE* by saying "the long violet tube that has two outlets on the side" versus the shorter "the long violet tube with two outlets on the side." My claim is that the speaker would use the relative clause version to emphasize the information in the relative clause. Relative clauses, thus, promote their contents (especially linguistically since they provide separation from the main clause) to an almost independent status. I feel this independent status stresses that the speaker took care in formulating the relative clause and that the information it conveys is less likely to be in error then if it had been expressed in a prepositional phrase or as an adjective; the water pump protocols tend to back up this claim (e.g., listeners would often use the information in a relative clause to confirm that their referent choice was correct). The head noun of the description can also be relaxed. It normally is relaxed last but could be relaxed prior to a relative clause (especially in the instances where the relative clause expresses confirmational information).

For example, consider the description "the blue cap that is on the main tube." Here, the features color and function are described in the adjective and head noun of the description, and the position in the relative clause. Following the rules suggested above, the relaxation of function and color should be attempted before position. The relaxation order proposed here is not meant to be the only way to relax the description. The order, in fact, may be modified by other knowledge sources.

5.1.3 Perceptual knowledge in reference

My system must take into account how people perceive objects in the world and how their perceptions can be represented. To do so, each object in the world has two representations in my system: a spatial (3-D) representation and a cognitive/linguistic representation that shows how the system could actually talk about the object. The spatial description is a physical description of the object in terms of its dimensions, the basic 3-D shapes composing it, and its physical features (along the lines developed in [2, 26]). It represents the result of human perceptual skill. The cognitive/linguistic form is a representation of the parts and features of the object in linguistic terms. In many ways this representation encodes the human capacity to extract information from our perceptual system and turn physical representations into words. It overlaps the spatial form - which holds relatively constant across people - in many respects but it is more suggestive of the listener's own perceptions. The cognitive/linguistic form often describes aspects of an object, such as its subparts, by its position on the object ("top", "bottom") and its functionality ("outlets", "places for attachment"). More than one cognitive/linguistic form can refer to the same physical description. Some properties of an object differ in how they are expressed in the two forms. In the 3-D form, there are primarily properties such as numerical dimensions (e.g., "3 feet by 5 feet") and basic shapes (e.g., generalized cylinders), while, in the cognitive/linguistic form, there are relative dimensions (e.g., "large") and analogical shapes (e.g., "the L-shaped tube").

Perceived objects, when spoken about, must be interpreted. This can lead to discrepancies between individuals. People usually agree on the spatial representation but not necessarily on the cognitive/linguistic description. This disagreement can lead to reference problems. For example, misjudgements by the speaker in calling an object "large" can cause the hearer to fail to find an object in the visual world that has dimensions that are perceptually "large" to the listener.

To avoid confusing the listener, a speaker must distinguish the objects in the environment from each other using perceptually useful features because these perceptual features provide people with a way to discriminate one object from another. A speaker must take care when selecting from these features since the hearer can become confused about the values of a feature irrespective of the actual object being described. Perceptual features may be inherently confusing because a feature's values are difficult to differentiate (e.g., is the tube a cylinder or a slightly tapering cone?). They may also be confusing because the speaker and listener may have differing sets of values for a feature (e.g., what may be blue for someone may be turquoise for another). These characteristics affect the salience of a feature (see [48] for a description of feature salience) which in turn determines the feature's usefulness in a description. A feature that is common in everyday usage (e.g., color, shape or size) is salient because the listener assumes that he can readily distinguish the feature's possible values from one another. Of course, very unusual values of a feature can stand out, making it even easier to discriminate a unique object from all other objects [48].

The objects in the world may exhibit a feature whose possible values are difficult to distinguish. This occurs when a perceived feature does not have much variability in its range of values: all the values are clustered closely together making it hard to tell the difference between one value and the next.[25] This increases the likelihood of confusion because the usefulness of specifying the feature to a non-expert is diminished (especially if the speaker is more expert than the listener in distinguishing feature values). Hence, if one of these difficult feature values appears in the

[25] For example, Burling [12, 16] contrasted vocabulary in Garo, a language spoken in Burma, with English. He found that some words in English were accounted for in Garo by many words. The world "rice" was represented in Garo by different names for "husked," "unhusked," "cooked," "uncooked," and other forms of rice. Such specialized names would be more difficult for non-Burmese to distinguish. Whorf [80] found similar results in his studies.

speaker's description, the listener, if he isn't an expert, will often relax the feature value to any of the members of the set of feature values. For example, if the speaker knows many shades of the color "red" (such as "scarlet," "crimson," "cherry," "maroon," or "magenta"), the average listener may not be able to distinguish them from each other and may be just as happy to pick up the "maroon plug" for the "magenta plug."

When the number of features available for describing an object is small, one could expect to have trouble discerning one object from the next depending on the quality of the features themselves. If the environment is full of objects whose perceived features (e.g., color, size or shape) are similar, one would expect more miscommunication the larger the similarities. In those cases where perceptual information can only group objects instead of highlighting a unique one, the members of the group might become distinguishable when functional information is added.[26] In other words, one may only know about the appearance of an object, but once one knows the function, the object and other potential contenders (might) become dissimilar [32].

5.1.4 Relaxing a description using perceptual knowledge

When examining the features presented in a speaker's description, one can consider perceptual aspects to determine which features are most likely in error. Such an inspection can generate a partial ordering of features for use during the repair process to determine which feature in a description to relax. As shown below, the relaxation ordering suggested by the inspection of features interacts with ordering proposals from other knowledge sources.

Active features are ones that require a listener to do more than simply recognize that a particular feature value belongs to a set of possible values – the

[26] Other descriptions such as "the second one from the left" are usable only when the speaker and listener are sharing the same perceptual view. Even when the same view is shared, the underlying task may also affect whether such a description is sufficient. For example, if the speaker is trying to teach an instructable robot how to perform a task, then a description such as "the second one from the left" may not be properly generalized by the robot for use in future perceptual views of the world.

listener must perform some kind of evaluation. They include the use of relative dimensions (e.g., "large"), comparatives (e.g., "larger") or superlatives (e.g., "largest"). When considering the water pump domain, I found that listeners were better at judging less active feature values (e.g., color values). Speakers, however, seem to be casual with less active features while the active ones require their full attention. Hence, in a reference failure the source of the problem is often the less active ones. This suggests that one should first relax those features that require less active consideration such as color (though it is easier to relax red to orange than red to blue), composition, transparency, shape and function because we would expect a speaker to be more serious about his use of active features. Only after this should one relax those features that require active consideration of the object under discussion and its surroundings (such as superlatives, comparatives, and relative values of size, length, height, thickness, position, distance and weight).

The water pump dialogues provided some evidence for this. For example, many speakers described the *MAIN-TUBE* using a relative size adjective such as "big" or "large." One of the descriptions of the tube was "the large blue tube." The *MAIN-TUBE* actually was violet but there was a blue tube, the *STAND*. Subjects still tended to select the *MAIN-TUBE* over the *STAND*, even with the color discrepancy, hinting that they preferred relaxing color (a less active feature) before relative size (an active feature).

5.1.5 Discourse knowledge in reference

Discourse knowledge concerns discourse structure, the flow of discourse and the use of discourse to highlight parts of the real world (see [30, 57, 69, 58, 4, 40, 56]for detailed treatments on discourse.). There are several mechanisms that can highlight objects in discourse (see work on focus by Grosz [30], Reichman [57] and Sidner [69]). They provide a partition of the real world that prunes the set of objects to consider during referent identification. Discourse knowledge also helps highlight what knowledge a speaker and listener have in common at any point in a dialogue. Conversants share knowledge about past actions and objects and general knowledge about the world (e.g., how to fit objects together or the functions of common objects). Focusing can demarcate which of several perspectives of world knowledge conversants should be using to interpret each other's utterances. This simplifies the amount of information that must be packaged in each utterance, reducing places for error. For example, deictics can be used to anchor descriptions to current or past context. The description "the yellow polka-dotted motor" requires a listener to look to see how the description hooks up to the current discourse situation. However, the description "the yellow polka-dotted motor I showed you yesterday" is anchored by the deictic "yesterday" and is more easily searchable.

5.1.6 Relaxing a description using discourse knowledge

Discourse knowledge helps the listener determine whether or not the problem is in the speaker's description or resides elsewhere. When normal reference fails (i.e., no referent corresponds to a description) and recovery is attempted, discourse knowledge can be used to determine whether the problem resides not in the description itself but possibly at the discourse level. For example, midstream corrections in an utterance by a speaker could cause a listener to either miss a shift in focus or to shift focus when no shift was intended. This was exemplified in Excerpt 6 in Chapter 2 when the speaker attempted to undo an earlier request and did not properly demark the shift of focus. The work of [30, 57, 78, 69, 32, 58] provided rules on deictics, anaphoric definite noun phrases, the use of pronominals versus nonpronominals, and so forth, that can be used to zero in on discourse problems. So, for example, if a self-correction of the use of a pronominal occurs (e.g., "...it - the X"), then a rule might state that focus could have shifted to X. Relaxation is then achieved by trying the hypothesized focus to see if a referent can now be found.

5.1.7 Hierarchical knowledge in reference

Imprecision (i.e., being overly general) in a speaker's description can lead to confusion. Being too specific can lead to similar results. Hierarchical knowledge - that is knowledge about a hierarchy of taxonomic information about our world - can be used by a listener to determine the degree of imprecision or specificity of a description. I can model this behavior by consulting a prestored generic/specific hierarchy of world elements, using the current context to guide the comparison of the speaker's current description to elements in the hierarchy, and deciding on the basis of the comparison if the description was imprecise.

An imprecise description, missing details needed to fully distinguish a real world object, should point out numerous candidates that exhibit the general features in the description rather than none at all. Imprecise descriptions can, however, lead to confusion that blocks the listener from finding any referent. If a feature is difficult to apply because it isn't specific or well-defined, then it may be necessary to ignore it (e.g., the use of a value like "funny" such as in "that funny red thing"). If a feature is ambiguous with respect to how it should be applied, then it may either require relaxation or further restriction (e.g., for the use of a feature value like "rounded," we must ask whether we mean "2-D" or "3-D" rounded? "cylindrical" or "bell-shaped"? and so on). The determination that a feature is too imprecise might be possible before a search for a referent is commenced. An examination of how high in the hierarchy the feature value appears could signal when a more detailed value is needed. Each of these problems was reflected in the water pump protocols by listeners. They often avoided searching for a referent because the speaker's description was just too imprecise, causing them confusion from the onset.

The condition of being too specific is more difficult to detect. In a task-oriented environment, one would not easily notice that something was too specific since normally being very specific is a wise goal for a speaker. The drawback of being too specific occurs not so much because of the specificity itself but because of its adverse side-effects. A description can be overspecific if it contains too many feature values or contains a feature that is overpowering. Section 2.3.4 describes these conditions in more detail.

5.1.8 Relaxing a description using hierarchical knowledge

Hierarchical knowledge can resolve certain ambiguities by climbing or descending the hierarchy. Such a hierarchy search requires looking at a description at two levels: (1) the description's placement in the generic/specific hierarchy and (2) the placement of the filler of each feature of the description in the generic/specific hierarchy.

Hierarchical knowledge also interacts with perceptual knowledge. The hearer can become confused when a feature value in the speaker's description is too hard to judge. For example, it is difficult to determine which particular feature value applies when the set of possible feature values are too specific. If a more imprecise value is used (and it applies only to one object), it might be easier to find the described object (e.g., "hippopotamus face shaped valve" would be better stated as "rounded valve"). Hence, in cases where a feature value is too specific, more imprecise values could be tried to see if a referent can then be found. These more imprecise values are found by looking higher in the hierarchy above the current feature value for more general terms.

5.1.9 Trial and error knowledge in reference

Trial and error knowledge has to do with performance feedback. Its primary use is to determine whether a referent was properly identified (including ones found with the relaxation process). Performance of a requested action is the strongest determining factor of whether or not the listener correctly interpreted a speaker's description.[27] Successful completion of an action will be likely to build confidence in the listener that he correctly interpreted a description. Failure to find an object after relaxation leads the listener to ask the speaker to clarify; failure to successfully perform the requested action on the object found during referent identification causes the listener to ask himself what is wrong. The trouble might be due to: (1) the object identified from the speaker's description, (2) the action attempted, or (3) some prior (probably unnoticed) mistake that occurred. Failure may come not only from the inability to perform an action but due to an action's postcondition failing.[28] Determination of how badly a postcondition must fail before the listener asks for clarification - instead of reconsidering the description - is unclear from the current protocols; further analysis collected from different protocols might resolve this matter.

[27] In more complex domains - such as ones requiring tools - the actions themselves may be helpful in both finding the referent and confirming whether the choice was correct. For example, if a listener is told to use a screwdriver to screw one object onto another, the listener would expect to find threads on the object.

[28] Note that the postcondition need not always be specified explicitly since some postconditions automatically come with an action. If the speaker said the utterance "fit *the red gizmo into the bottom side outlet of the main tube,"* the listener would expect that the red gizmo would fit snugly into the outlet. If, however, it fit loosely, than the listener may feel a mistake has occurred.

5.2 Representation of the knowledge sources for rule based relaxation

This section formalizes some of the knowledge sources described in the previous section. The basic mechanism is a set of rules that drive the relaxation process. The rules detect reference miscommunication, order the features in a speaker's description for relaxation, relax the speaker's description to fit the best referent candidate, and determine if the selected referent is correct.

5.2.1 Rules for handling miscommunication

The purpose of these rules is to recognize trouble before, during, or after the search for a referent. This section provides a sampling of the kinds of rules that were developed.

5.2.1.1 Before search for referent

A listener can detect trouble with a speaker's description before searching for a referent if the description contains imprecise features or uses a feature value that is too specific. The use of imprecise feature values without some precise ones to counter them, or the use of feature values that are too specific, strongly suggest that the listener avoid the actual search for a referent. An attempt to judge the imprecision and specificity of a feature can be done using hierarchical knowledge.

Hierarchical knowledge provides a taxonomy of features and their possible values. Some features are very precise (e.g., "globe-shaped," "spherical," or "hippopotamus-shaped") while others are imprecise (e.g., "rounded"). The taxonomy distinguishes precise terms from imprecise ones by placing the precise ones lower in the taxonomy. One way of predicting whether or not a feature value in a description is imprecise is to compare the number of concepts above and below it in the taxonomy. The current heuristic used is that a concept in the taxonomy is imprecise if the longest path of concepts from the concept to the bottom of the taxonomy exceeds, by at least one, the longest path of concepts from the concept to the top of the taxonomy. This is meant as an operational definition and not intended to represent any cognitive aspects of imprecision.

A cognitive definition of imprecision might be based on the results of category theory and the use of basic categories (as defined by Rosch in [64]). A basic category characterizes the terms people use in their daily life. Imprecision of a feature value could be defined by looking at where it sits in the basic category to which it belongs. One, thus, can tell if a feature value is outside the norm or not. The boundaries would be fuzzy but three sets of feature values could be distinguished from a basic category description of a feature: (1) those more specific than the usual feature values used in the basic category, (2) those less specific than the normal ones, or (3) those values normally used by people to describe terms in the basic category. Likewise, a feature value is too specific if it has lots of concepts above it in our taxonomy of features and feature values. Figure 5-1 shows that "round" is imprecise while "hippopotamus-shaped" is very precise. Note how both definitions of imprecision given above are intended to prevent "red" from being described as too precise while allowing for "hippopotamus-shaped" to be overly precise. A concept, thus, is not too specific just because it is the lowest concept in the taxonomy for a particular physical property. It is also important to consider how many concepts exist between the most specific concept and the least specific concept of that physical property.

A feature value's position in the taxonomy also is not always a clear indicator of a feature value's imprecision or specificity because the physical context has an influence. For example, "rounded" may be a perfectly reasonable way to refer to a cylinder that is on a table containing one cylinder and a bunch of blocks. There are also feature values that are imprecise no matter where they appear in the taxonomy. For example, consider the use of a feature normally applied to an animate object to describe an inanimate object. A description like "the pretty one" is not readily applicable to an inanimate object (though once a speaker identifies an object as exhibiting such a feature, an anaphoric reference using the feature is alright). Such feature values can be marked in the taxonomy as special cases or the taxonomy itself can be forced to provide such information by carefully splitting properties of animate and inanimate objects into those that are applicable only to animate, only to inanimate, or to both.

A sample of the rules relevant before the referent search commences are shown below. The rules are described as situation recognition rules, i.e., a pattern is presented and, if the pattern holds, a particular situation is recognized.

a,b,c <- x,y,z

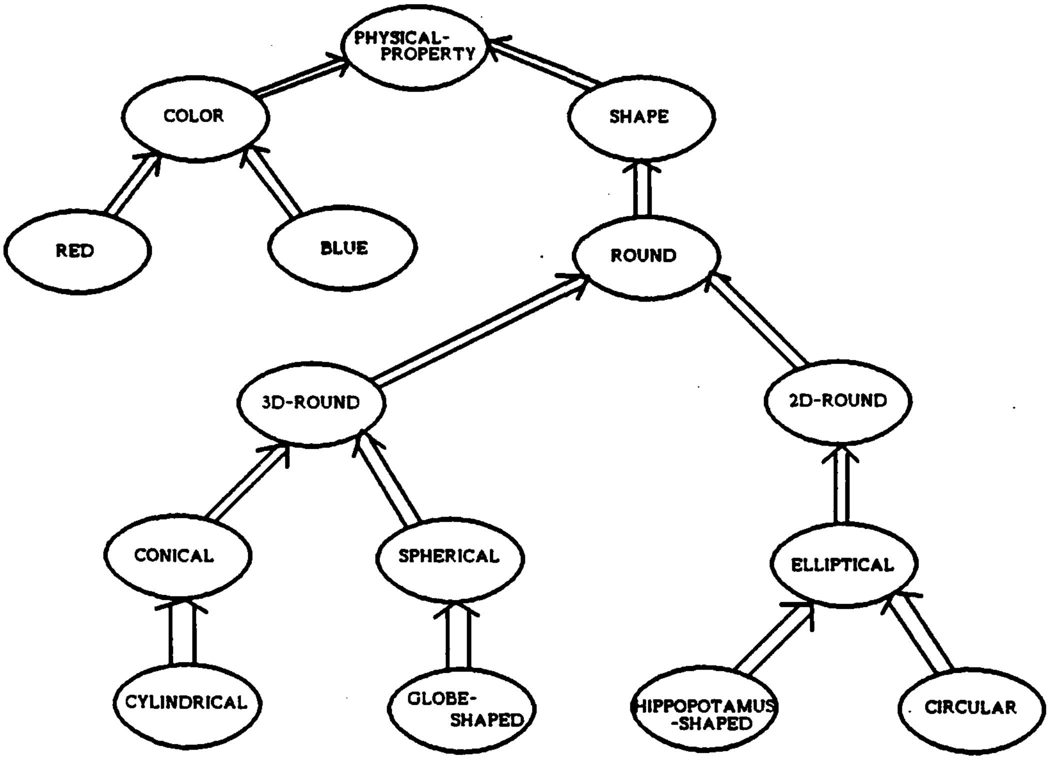


Figure 5-1: *Hierarchy of feature values*

The pattern part, which is the right-hand side of the rule, is separated from the situation part, which is the left-hand side of the rule, by an arrow ("<-"). The pattern part is composed of a list of predicates and functions that must all be satisfied before the situation is recognized. Each predicate is separated from another by placing a comma in between. All predicate names begin with an uppercase character and all function names are in lowercase. The situation part tells what disjunction of situations holds if the pattern part is true. The whole rule is equivalent, in logical notation, to

x AND y AND z ==> a OR b OR c.

When there is nothing in the pattern side of the rule, then the situation side is asserted to be true (i.e., "a OR b OR c" is true).

a,b,c <-

If there is nothing in the situation part of the rule, then whatever is in the pattern side is asserted to be false (i.e., "x AND y AND z" is false).

<- x,y,z

This notation is similar to that used for Horn clauses.

The values of the arguments used by the predicates and functions below come from the speaker's description. The previous chapter described how the parser and semantic interpreter generate a representation of the speaker's description in KL-One. The KL-One representation contains a set of features and feature values. These values can be retrieved using basic KL-One functions. For example, the value of the COLOR role on a description can be retrieved to get the color value specified by the speaker (e.g., the function KLFindValueDescriptions[|R|COLOR;|C|DESCR] would return the value of the COLOR role on the concept DESCR.).

Applicable predicates and functions

getallfeatures[d] This function retrieves the names of all the features that are present in the KL-One representation of the speaker's description, d. It returns the features in a list.

getallfeaturevalues[d]

This function retrieves the feature values of all the features that are present in the KL-One representation of the speaker's description, d. It returns the values in a list.

getfeaturevalue[d,f]

This function retrieves the feature value of feature f in the KL-One representation of the speaker's description, d.

ObjectDescr[d] This predicate is true if its argument, d, is a description from the speaker that is meant to refer to some object in the world.

Utterance[u] This predicate is true if its argument, u, is an utterance from the speaker.

VagueFeature[v] This predicate determines whether or not the feature value, v, is imprecise.

It searches the taxonomy checking the feature value's position to see if it is high in the taxonomy. The current heuristic used is that a concept in the taxonomy is imprecise if the longest path of concepts from the concept to the bottom of the taxonomy exceeds, by at least one, the longest path of concepts from the concept to the top of the taxonomy.

It checks if the feature value is not easily applied to the objects in the domain.

AllFeaturesVague[d]

This predicate checks to see if all the feature values in a description, d, are imprecise.

```
AllFeaturesVague(NIL)<-
AllFeaturesVague(getallfeaturevalues(d))
      <- ObjectDescr(d),
         VagueFeature(car (getallfeaturevalues(d))),
         AllFeaturesVague(cdr (getallfeaturevalues(d)))
```

VerySpecificFeature[v]

This predicate determines whether or not a feature value, v, is very specific. It searches the taxonomy checking the feature value's position to see if it is low in the taxonomy.

DescrWithVerySpecificFeature[d]

This predicate determines if a description, d, contains a very specific feature value.

```
DescrWithVerySpecificFeature(getallfeaturevalues(d))
    <- ObjectDescr(d),
       VerySpecificFeature(car (getallfeaturevalues(d)))
DescrWithVerySpecificFeature (getallfeaturevalues(d))
    <- ObjectDescr(d),
       DescrWithVerySpecificFeature
                            (cdr (getallfeaturevalues(d)))
```

Sample rules

```
AvoidSearchForReferent(x)
        <- ObjectDescr(x),AllFeaturesVague(x)
```

If the above rule is true, a listener should ask the speaker for more information before looking for a referent.

```
AvoidSearchForReferent(x)
        <- ObjectDescr(x),DescrWithVerySpecificFeature(x)
```

If the above rule is true, a listener can ask the speaker for more information, attempt to use a less *specific feature* value (e.g., substitute "rounded" for "hippopotamus-shaped"), or ignore the very precise feature value altogether.

5.2.1.2 During search for referent

A listener can detect confusion on the part of the speaker during the search for a referent if the speaker interrupts his own utterance.[29] An interruption can come about with a false start or a self-correction. A false start occurs when the speaker goofs on his initial description, stops, and then restarts the description. For example, exclamations like "oops," "never mind," "oh no," and so on are signals of false starts meant to inform the listener that there is a problem, though not stating precisely where the problem occurred. The problem could be due to the current utterance or a previous one. Speaker's often (falsely) assume the listener "knows" just where the speaker means. Typically, a listener presumes the problem is with the current utterance. A listener should, however, note that a false start has occurred at this point in the dialogue and be prepared to back up to the same place later on. Self-corrections are less interruptive than false starts and more explicit about the source of the problem. They are redescriptions of a piece of the speaker's utterance that occur as it is spoken. Descriptions like "it--the tube" or "the large blue--uh violet tube" are typical ones that occur. As with false starts, such places are conducive to confusion and should be noted by the listener.

Another problem occurs when a speaker expresses one value for a feature in a description and then, later on in the same description, contradicts that feature value

[29] These interruptions are more typical of spoken rather than written language.

by giving another one. This might be a feeble attempt at self-correction by the speaker or a way to vaguely define an unnamed or unknown feature value. For example, the description "the plastic cylinder that is made out of metal" or "the red tube that is yellow" seem contradictory while "the blue-green cylinder" may not since it could be referring to a color like turquoise. These feature values blatantly contradict each other, often leading to confusion on the part of the listener. Such descriptions should be noted immediately by the listener as a problem. The listener can ask the speaker for clarification or consider ignoring the contradictory feature values.

Applicable predicates and functions

FalseStart[u] This predicate determines whether or not a false start has occurred in some utterance, u. Such false starts would have to be caught by the parser.

Self-Correction[d]

This predicate looks for self-corrections in a description, d. As with FalseStart, it would have to be implemented inside the parser.

feature[v] This is a function that returns the feature of the feature value, v, passed to it as an argument.

Feature[f] This predicate is true if its argument, f, is a physical feature.

FeatureDescriptor[v]

This predicate determines if its argument, v, is a feature value of any of the features.

FeatureInDescription[v,d]

This predicate determines if a feature value, v, is contained in a description, d.

BlatantlyContradictoryFeatureValues[v1,v2]

This predicate checks to see if, in the same description, two different feature values are given for the same feature.

```
BlatantlyContradictoryFeatureValues(v1,v2)
        <- FeatureDescriptor(v1),FeatureDescriptor(v2),
           Equal(feature(v1),feature(v2)),
           Not(Equal(v1,v2))
```

MarkForPossibleConfusion[u]

This predicate is true when something appears confused in the description or utterance, u. It is used to mark u as having a possible problem that may need to be checked further.

Sample rules

```
MarkForPossibleConfusion(u)
        <- Utterance(u),FalseStart(u)

MarkForPossibleConfusion(d)
        <- ObjectDescr(d),Self-Correction(d)

MarkForPossibleConfusion(d)
        <- ObjectDescr(d),FeatureInDescription(v1,d),
           FeatureInDescription(v2,d),
           BlatantlyContradictoryFeatureValues(v1,v2)
```

The above rules mark the utterance u as a possible place to back up to should confusion occur later on in the dialogue.

5.2.1.3 After search for referent

The results of a search for a referent tells us whether the search succeeded or not and, if not, can hint at the general kind of problem that has occurred. The referent is successfully found; more than one referent, when only one was expected, is found; a referent is found but the action to perform on it fails in some way; or no referent is found. The first result implies that there is no problem, but the others indicate trouble. The second one means that the speaker's description is ambiguous. That result implies (1) one or more of the feature values specified in the speaker's description is not precise enough or (2) too few features are presented in the speaker's description. Both problems require the listener to get clarification from the speaker; otherwise, the listener will have to try each of the ambiguous objects to see if the action requested to perform on the part actually succeeds. The third possible result of the referent search indicates that something is wrong with the speaker's description of either the object or the action. In either case, the listener requires clarification from the speaker to resolve the problem. The listener, sometimes, will assume that the failure of the action indicates that the referent he found was wrong. In that case, since no other referent exisits, the listener would enter the fourth category - the one where no referent is found. This category is the most interesting of all. Since no referent was found, then something is probably wrong with the speaker's description. This leads to the relaxation of the speaker's description by the listener.

A speaker's description is often relaxed in an orderly manner, using the knowledge that a listener has about his world. The first part of this chapter described numerous kinds of knowledge that people use to search for referents and to recover from mistakes that occur. Much of that knowledge can be used to order parts of a speaker's description for relaxation during the repair of the description. Below I attempt to formalize some of that knowledge using the rule format I introduced earlier in this section.

Rules for ordering features for relaxation

Applicable predicates and functions

syntactic-form[v,d]

This function returns the kind of syntactic category (ADJective, Prepositional Phrase, RELative CLauSe, or PREDicate COMPlement) used in the speaker's description, d, to describe a feature value, v.

World[w] This predicate is used to indicate a particular world, w, in which something holds true. The world can be a particular domain (e.g., the water pump task), a particular person and so forth.

WorldObject[o,w] This predicate is used to determine if an object, o, is part of some particular world, w.

Superlative[v] This predicate is true if its argument is a feature value, v, that is expressed as a superlative (e.g., "largest").

Comparative[v] This predicate is true if its argument is a feature value, v, that is expressed as a comparative (e.g., "larger").

Relative-Feature[v]

This predicate is true if its argument is a feature value, v, that is expressed as a relative dimension (e.g., "large").

ColorValue[c] This predicate determines whether or not its argument, c, is a kind of color.

Color[c,o] This predicate is true if its first argument, c, is the color of the object represented by the second argument, o.

Similar-Color[c1,c2]

This predicate is true if its two arguments, c1 and c2, are both color values that are very similar in color.

ActiveFeature[v] This predicate is true if its argument, v, is a feature value of one of the "active" features (e.g., a relative dimension such as size, a comparative, or a superlative).

NonActiveFeature[v]

This predicate is true if its argument, v, is a feature value of one of the "non-active" features (e.g., color, shape, transparency).

ClusteredFeatureValues[f,w]

This predicate is true if one of its arguments, a physical feature f, is composed of lots of feature values that are close to each other in the world defined by its second argument, w.

FitCondition[c] This predicate is true if its argument, c, is one of "TIGHT," "LOOSE," "NO-FIT," "VERY-LOOSE," or "FITS-OK." Each of those is meant to describe how well two objects fit together.

Relax-Feature-Before[v1,v2]

This predicate is true if feature value v1 should be relaxed before feature value v2.

Linguistic knowledge rules and assertions

```
Equal(syntactic-form(v,d),"RELCLS"),
Equal(syntactic-form(v,d),"PP"),
Equal(syntactic-form(v,d),"ADJ"),
Equal(syntactic-form(v,d),"PREDCOMP")
  <- FeatureDescriptor(v),ObjectDescr(d),FeatureInDescription(v,d)
```

The above rule asserts that each feature value of a speaker's description can be specified as one of the syntactic forms: relative clause, prepositional phrase, adjective or predicate complement.

```
Relax-Feature-Before(v1,v2)
  <- ObjectDescr(d),FeatureDescriptor(v1),FeatureDescriptor(v2),
     FeatureInDescription(v1,d),FeatureInDescription(v2,d),
     Equal(syntactic-form(v1,d),"ADJ"),
     Equal(syntactic-form(v2,d),"PP")
```

Relax a feature value specified as an adjective before one specified as a prepositional phrase.

```
Relax-Feature-Before(v1,v2)
  <-ObjectDescr(d),FeatureDescriptor(v1),FeatureDescriptor(v2),
     FeatureInDescription(v1,d),FeatureInDescription(v2,d),
     Equal(syntactic-form(v1,d),"ADJ"),
     Equal(syntactic-form(v2,d),"RELCLS")
```

Relax a feature value specified as an adjective before one specified as a relative clause.

```
Relax-Feature-Before(v1,v2)
```

```
<-ObjectDescr(d),FeatureDescriptor(v1),FeatureDescriptor(v2),
   FeatureInDescription(v1,d),FeatureInDescription(v2,d),
   Equal(syntactic-form(v1,d),"ADJ"),
   Equal(syntactic-form(v2,d),"PREDCOMP")
```

Relax a feature value specified as an adjective before one specified as a predicate complement.

The above rules provide only a sample of the possible linguistic knowledge ordering rules.

Perceptual knowledge rules and assertions

```
Similar-Color("RED","PINK")<-

Similar-Color("RED","MAROON")<-

Similar-Color("GREEN","EMERALD")<-

Similar-Color("GREEN","BLUE-GREEN")<-

Similar-Color("BLUE","NAVY-BLUE")<-

Similar-Color("BLUE","TURQUOISE")<-
          .
          .
          .

where Color("RED"), Color("PINK"), Color("MAROON"), Color("GREEN"),
      Color("EMERALD"), Color("BLUE"), Color("BLUE-GREEN"),
      Color("NAVY-BLUE"), and Color("TURQUOISE").
```

The above are a sample of assertions specific to a particular domain (and a particular person). Here the assertions describe some of the colors in the world that are similar to each other. Corresponding assertions exist for other physical properties such as transparency or shape. These assertions are used by the relaxation mechanism to determine if it is reasonable to substitute one value for another.

```
ActiveFeature(v)<- FeatureDescriptor(v),Superlative(v)

ActiveFeature(v)<- FeatureDescriptor(v),Comparative(v)

ActiveFeature(v)<- FeatureDescriptor(v),Relative-Feature(v)
```

Active features include superlatives, comparatives or relative features that require evaluation on the part of the listener.

```
Relax-Feature-Before(v1,v2)
  <- ObjectDescr(d),FeatureDescriptor(v1),FeatureDescriptor(v2),
     FeatureInDescription(v1,d),FeatureInDescription(v2,d),
     NonActiveFeature(v1),ActiveFeature(v2)
```

Relax less active features before active features.

```
ClusteredColorValues(w)
  <- Feature(COLOR),World(w),
     ColorValue(c1),ColorValue(c2),ColorValue(c3),
     WorldObject(o1,w),WorldObject(o2,w),WorldObject(o3,w),
     Color(c1,o1),Color(c2,o2),Color(c3,o3),
     Similar-Color(c1,c2),Similar-Color(c1,c3),
     Similar-Color(c2,c3)
```

A world may contain clustered values of a physical feature. The feature may have possible feature values that are spread all over the spectrum but, for the current world view, many of the objects exhibit values that are all very near each other and, thus, hard to distinguish. The above rule defines "clustered color values" as meaning that the physical world under consideration has three or more objects that have similar colors. It is meant as an exemplar for a whole series of rules (e.g., ClusteredShapeValues, ClusteredTransparencyValues and so on).

```
Relax-Feature-Before(v1,v2)
        <-ClusteredFeatureValues(feature(v1),w),
          NOT(ClusteredFeatureValues(feature(v2),w))
```

The above rule says to relax a feature value of a clustered feature before one of a non-clustered feature.

The above rules are meant to be suggestive of the kinds of rules one can write to represent perceptual knowledge to use in the relaxation process.

Trial and error rules and assertions

The set of assertions that follow are meant to simulate the result of fitting two objects together in the water pump world. The predicate used is FitP(o1,o2,c) where WorldObject(o1), WorldObject(o2), and FitCondition(c).

```
FitP(TUBEBASE,THREADED-ENDofMAIN-TUBE,"TIGHT")<-

FitP(TUBEBASE,UNTHREADED-ENDofMAIN-TUBE,"LOOSE")<-

FitP(SLIDEVALVE,HOLEofTUBEBASE,"FITS-OK")<-
FitP(SLIDEVALVE,OUTLET1ofMAIN-TUBE,"VERY-LOOSE")<-

FitP(SLIDEVALVE,OUTLET2ofMAIN-TUBE,"LOOSE")<-

FitP(SLIDEVALVE,BottomHOLEofSTAND,"VERY-LOOSE")<-

FitP(SLIDEVALVE,TopHOLEofSTAND,"VERY-LOOSE")<-
   .
   .
   .
FitP(LargeENDofSPOUT,BottomofTUBEBASE,"LOOSE")<-

FitP(SmallENDofSPOUT,BottomofTUBEBASE,"FITS-OK")<-

FitP(SmallENDofSPOUT,OUTLET1ofMAIN-TUBE,"TIGHT")<-

FitP(SmallENDofSPOUT,OUTLET2ofMAIN-TUBE,"TIGHT")<-
   .
   .
   .
```

The following rules express when a listener realizes something might be wrong because the fitting together of two objects yields a fit condition different from the one either expressed to the listener by the speaker (the first rule) or expected by default (the second rule).

```
MarkForPossibleConfusion(d)
  <- ObjectDescr(d),FitCondition(c1),FeatureDescriptor(c1),
     FeatureInDescription(c1,d),FitCondition(c2),
     Not(Equal(c1,c2))

MarkForPossibleConfusion(d)
  <- ObjectDescr(d),FitCondition(c1),
     Not(Equal(c1,"FITS-OK"))
```

5.3 Summary

This chapter demonstrated that recovery from reference failure uses broad forms of knowledge about language and the physical world around us. These knowledge sources provide us with heuristics - represented here as relaxation rules - for coping with poor or errorful descriptions. I showed how the rules could predict where the problems were in a speaker's descriptions. I neglected to describe how such rules can actually be used and ignored the fact that the rules can conflict with each other. The next chapter describes a control structure for using the rules.

6. THE RELAXATION COMPONENT

6.1 Introduction

I discussed in the previous chapter some of the numerous kinds of knowledge available to a listener to interpret a speaker's description. I pointed out places where that knowledge affects the listener's ability to interpret a description and ways in which it is helpful to the listener for overcoming poor descriptions. When a description fails to denote a referent in the real world properly, it is possible to repair it by a relaxation process that ignores or modifies parts of the description. Since a description can specify many features of an object, the order in which parts of it are relaxed is crucial (i.e., relaxing in different orders could yield matches to different objects). There are several kinds of relaxation possible. One can ignore a constituent, replace it with a related value, or change focus (i.e., consider a different group of objects). This chapter describes the overall relaxation component that draws on the knowledge sources about descriptions and the real world as it tries to relax an errorful description to one for which a referent can be identified.

6.2 Find a referent using a reference mechanism

Identifying the referent of a description requires finding an element in the world that corresponds to the speaker's description (where every feature specified in the description is present in the element in the world but not necessarily vice versa). This process corresponds to the technique employed in the traditional reference mechanisms. The initial task of my reference mechanism is to determine whether or not a search of the linguistic world and real world knowledge base is necessary. For example, in the water pump domain, the reference component should not bother searching – unless specifically requested to do so – for a referent for indefinite noun phrases (which usually describe new or hypothetical objects) or extremely vague descriptions (which are ambiguous because they do not clearly describe an object since they are composed of imprecise feature values). A noun phrase can be determined by the parser as indefinite or not. There was a rule suggested in the previous chapter for determining whether or not a description is imprecise. A number of aspects of discourse pragmatics can also be used in determining whether or not to search for a referent. For example, the use of a deictic in a definite noun phrase, such as "this X" or "the last X," hints that the object was either mentioned previously or that it probably was evoked by some previous reference, and that it is searchable. I will not examine such aspects any further in this thesis since my main interest is in recovery from failures of reference that occur during or after the search for a referent.

Once a search of the knowledge base is considered necessary, a reference search mechanism is invoked. As I described in Chapter 4, the search mechanism uses the KL-One Classifier [39] to search the knowledge base taxonomy. This search is constrained by the focus mechanism described in Chapter 4. Descriptions of possible referents of the speaker's description will be subsumed by the description after it has been classified into the knowledge base taxonomy. If more than one candidate referent is below the classified description, then, unless a quantifier in the description specified more than one concept, the speaker's description is ambiguous. If exactly one concept is below it, then the intended referent is assumed to have been found. Finally, if no referent is found below the classified description, the relaxation component can be invoked. Prior to actually using the relaxation component, FWIM checks to see if the problem resides not with the description but due to pragmatic issues. I will only consider the no referent case in the rest of this chapter.

6.3 Collect votes for or against relaxing the description

If the referent search fails, then it is necessary to determine whether the lack of a referent for a description has to do with the description itself (i.e., reference failure) or outside forces that are causing reference confusion. For example, an external problem due to outside forces may be with the flow of the conversation and the speaker's and listener's perspectives on it; it may be due to incorrect attachment of a modifier; it may be due to the action requested; and so on. Pragmatic rules are invoked to decide whether or not the description should be relaxed. Some of these rules were described in the last chapter in Section 5.2.1.2. For example, misfocus, which can lead to the speaker and listener having different perspectives on the current focus of attention, is detected by the rules on false starts and self-corrections. If the rules indicate the likelihood of misfocus, then the speaker's description should not be relaxed and a referent should be looked for in another part of the real world in a different focus space. Other pragmatic rules deal with such issues as metonomy and synecdoche. These rules will not be discussed here; we will assume that the problem lies in the speaker's description.

6.4 Perform the relaxation of the description

If relaxation is demanded, then the system must (1) find potential referent candidates, (2) determine which features in the speaker's description to relax and in what order, and use those ordered features to order the potential candidates with respect to the preferred ordering of features, and (3) determine the proper relaxation techniques to use and apply them to the description.

6.4.1 Find potential referent candidates

Before relaxation takes place, the algorithm looks for potential candidates for referents (which denote elements in the listener's visual scene). These candidates are discovered by performing a "walk" in the knowledge base taxonomy in the general vicinity of the speaker's classified description as partitioned by the focusing mechanism.[30] A KL-One partial matcher, which is described in more detail in Section 6.4.1.1, is used to determine how close the candidate descriptions found during the walk are to the speaker's description. The partial matcher generates a numerical score to represent how well the descriptions match (after first generating scores at the feature level to help determine how the features are to be aligned and how well they match). This score is based on information about KL-One (e.g., the subsumption relationship between or the equality of two feature values) and does not take into account any information about the task domain. The set of best descriptions returned by the matcher (as determined by some cutoff score) is selected as the set of referent candidates. The ordering of features and candidates for relaxation described in Section 6.4.2 below takes into account the task domain.

[30] Appendices E and F show example walks in a knowledge base taxonomy.

For the moment, the implemented exploration routine explores the entire taxonomy checking to see if each concept would be a reasonable match to the speaker's description. It is really unnecessary, however, to explore the whole taxonomy. There are several ways one can reduce the amount of searching. One can stop checking above a particular concept if the score between the current concept and the speaker's description is too low. The exploration routine would still, however, check other concepts below the current concept since those concepts might better match the speaker's description. I mentioned in the last chapter how basic categories [64] can be used to determine in a hierarchy whether a description is imprecise or not. Basic categories can also be used to prune the search space of the exploration routine. They can form non-overlapping parts of the hierarchy (e.g., dividing the hierarchy into different types of objects and physical relations). The walk in the taxonomy can avoid searching concepts, and all their descendants, if they are in a basic category different than that of the original description. There is one problem with using basic categories in that manner. Since the relaxation component is dealing with descriptions that have mistakes in them, it is possible that a concept in a different basic category is really the correct concept.

6.4.1.1 Perform a partial match of two KL-One descriptions

The taxonomy walk described above will yield a potentially large group of candidate referents. The KL-One partial matcher that I implemented is used to reduce that set down to a manageable number of referent candidates. This section describes how the partial matcher works.

The matching of two KL-One descriptions requires determining how the concepts are related in the taxonomy and performing an alignment of the roles on one concept to those on the other. Both tasks are achieved by the KL-One partial matcher by taking advantage of the inherent structure of KL-One descriptions. Each concept in the taxonomy is related in at least one of the following ways to every other concept: (1) one concept subsumes or is subsumed by the other, (2) both concepts are subsumed by another, non-root, concept, or (3) both concepts are subsumed by the root concept (THING). Cases (1) and (2) make it easier to match the two concepts because role alignment is usually simpler. Case (2) actually includes case (1) since nothing prevents the subsumer concept from being one of the two concepts. For that reason, I will concentrate primarily on cases (2) and (3).

Role alignment attempts to (uniquely) align a role of the first concept with a corresponding role on the second. The roles are not aligned by name because KL-One ignores the name of a role (as well as names of concepts) since there isn't always an exact correspondence between roles names. For example, the role ARM could appear on one concept and the role LIMB could appear on the other concept, yet both may be reasonable to align with each other. When two concepts are related through subsumption, then all the roles on the subsumer (the concept higher in the taxonomy) are also represented on the subsumee concept (the lower concept). Each subsumer role appears on the subsumee concept exactly as it does on the subsumer concept or it can be "modified" or "differentiated" slightly into a more specific role.[31] If the role on the subsumee is modified, then a "Mods" link runs from it to the corresponding role on the subsumer concept. Similarly, for differentiation, a "Diffs" link runs from the subsumee role to the corresponding subsumer role. These role links are important for role alignment because they provide strong evidence that two roles are related. In Figure 6-1, role R3 on concept A is related to role R1 on concept C since a series of "Mods" links can be followed from role R3 to role R1. If a sequence of "Mods" and "Diffs" links also exists from role R4 on concept B to role R1, then role R3 and role R4 are very likely candidates for alignment to each other. Similarly, roles R5 and R7 on concept A have a "Diffs" link to role R2 on concept C while role R6 on concept B also has a "Diffs" link to R2. Hence, roles R5 and R7 will likely be aligned to role R6. The actual partial matcher procedure works by chasing up all "Mods" and "Diffs" links on a role until it can go no higher. It then compares the path of role links generated for the role on one concept to the path of role links generated for a role on another concept. If the two paths intersect somewhere, then there is strong evidence that the two roles should be aligned. If no such intersection occurs, then a second level of comparison between roles is required.

Two roles can also be compared by examining their value restrictions, with the following possible results: the two value restrictions are the same; one value restriction subsumes the other; both value restrictions share a common, non-root, subsumer; or no direct relation is seen between the two value restrictions. Each of

[31] These terms were defined in Section 4.1. They refer to role modification and role differentiation.

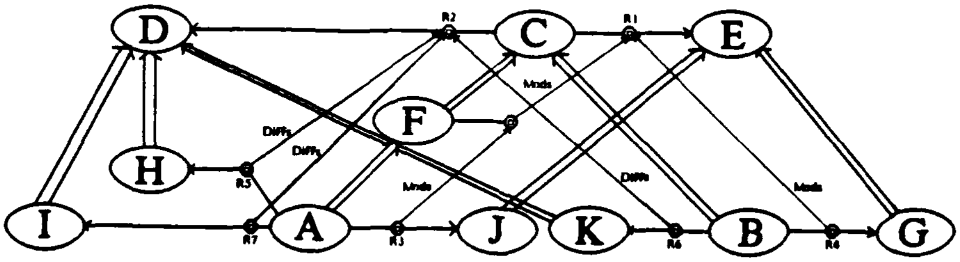


Figure 6-1: Aligning roles

those respective cases is weaker than the one before it. In the last two cases, the partial matcher is called recursively to determine how well the two value restrictions match. This recursive matching will eventually terminate when the value restriction of a role is a primitive concept or it will cycle and repeat an earlier value restriction (at which point the matching procedure stops).

A numerical score is generated for each role alignment pair. The score is based on whether or not there is an intersection along the path of role links generated for each role, how the value restrictions of each role are related, whether or not the name of each role is the same, and whether the number restrictions of each role are consistent with each other. An overall score for the concept match is generated from the role alignment scores and from a distance measure that determines how far apart "conceptually" the two concepts are in the taxonomy. Since there are potential conflicts between role alignments (e.g., two roles on one concept may align to the same role on another concept), the overall score is actually given as a range of values. This range is specified as a pair of scores. The first number provides the lowest possible overall score that is calculated from the set of feasible role alignments while the second number provides the highest possible score.

6.4.2 Order the features and candidates for relaxation

At this point the reference system inspects the speaker's description and the candidates, decides which features to relax and in what order,[32] and generates a master ordering of features for relaxation. Once the feature order is created, the reference system uses that ordering to determine the order in which to try relaxing the candidates.

The various knowledge sources described in the previous chapter are consulted to determine the feature orderings for relaxation. Each knowledge source produces its own partial ordering of features using the set of relaxation rules defined in Chapter 5. An example of one of these rules for linguistic knowledge is shown in Figure 6-2. The partial orderings are then integrated to form a directed graph. For example, perceptual knowledge may say to relax color. However, if the color value was asserted in a relative clause, linguistic knowledge would rank color lower, i.e., placing it later in the list of things to relax.

Relax the features in the speaker's description in the order: adjectives, then prepositional phrases, and finally relative clauses and predicate complements.

```
E.g.,
  Relax-Feature-Before(v1,v2)
    ←ObjectDescr(d),FeatureDescriptor(v1),
      FeatureDescriptor(v2),
      FeatureInDescription(v1,d),
      FeatureInDescription(v2,d),
      Equal(syntactic-form(v1,d),"ADJ"),
      Equal(syntactic-form(v2,d),"REL-CLS")
```

Figure 6-2: A sample relaxation rule

Since different knowledge sources generally produce different partial orderings of features, these differences can lead to a conflict over which features to relax. It is the job of the best candidate algorithm to resolve these disagreements among knowledge sources. Its goal is to order the referent candidates, C_1, C_2, ..., C_n, so that relaxation is attempted on the best candidates first. Those candidates are the ones that conform best to a proposed feature ordering.

[32] Of course, once one particular candidate is selected, then deciding which features to relax is relatively trivial - one simply compares feature by feature between the candidate description (the target) and the speaker's description (the pattern), and notes any discrepancies.

```
Set of feature orderings:  $F_1, F_2, \ldots, F_n$
                           where each $F_i$ is an ordered set
                           of features $\{f_1, f_2, \ldots, f_m\}$

Speaker's description:  $D=\{v_1, v_2, \ldots, v_l\}$
                        where each $v_i$ is a feature value
                        specified in the speaker's description.

Set of referent candidates:  $C_1, C_2, \ldots, C_k$

For each pair $(C_i, F_j)$,
 where i runs from 1 to k and
       j runs from 1 to n,
         generate a score that represents the consequence
           of relaxing description D to candidate $C_i$
           using feature ordering $F_j$.

Scoring the pair $(C_i, F_j)$:

  (1) Penalize more the score of those pairs which require
      relaxing feature values of D whose corresponding
      features occur farther into the feature ordering $F_j$.

        Assign an integer that corresponds to the position in
         $F_i$ that represents the feature that must be be
         relaxed in D to match $C_i$.  It ranges from 1,
         which is the first position in $F_j$, to m, which
         is the last position in $F_j$.  It is 0 when no
         features are relaxed.

        Repeat for each feature that must be relaxed in D.

        Sum up all the position numbers for relaxed features
          and assign to SUM.

  (2) Penalize the score of those pairs which require relaxing
      more feature values of D.

        Assign a number based on the difference between the
         number of feature values in D and the number of
         features that must be relaxed.  Store that number
         in #NOTRELAXED.  It can range from 0 to l.

  Then, $SCORE(C_i, F_j)$ <- SUM/#NOTRELAXED.
```

Figure 6-3: Choosing the "best" referent candidates

```
Scoring each $C_i$:
  Generate a score that tells how well candidate $C_i$
    conforms to the feature orderings $F_1,F_2,...,F_n$
    when applied to description D.
  $SCORE(C_i) <- SUM[SCORE(C_i,F_j)]$ for j=1 to n.
  Reorder the referent candidates, $C_i$, so that those
    with the least scores go first.
```

FIGURE 6-3, CONCLUDED

Figure 6-3 sketches one possible algorithm for choosing and ordering the best referent candidates using the directed feature order graph. Since the number of possible referent candidates has been reduced by the partial matcher, the number of features is small, and the number of paths through the graph is reasonably limited, it is reasonable to try all combinations of proposed feature relaxation orderings on each referent candidate.[33] The algorithm determines that the best candidates are the ones that both minimize the number of features relaxed and require the relaxation of features found "earliest" in the feature ordering. This criterion ignores the actual feature values themselves and how reasonable it is to relax a particular feature value in the speaker's description to the one exhibited by the referent candidate (e.g., is it alright to relax "blue" to "red?"). The feature values are considered in another phase of the relaxation algorithm described in Section 6.4.3.

The goal of the algorithm is to order the referent candidates, C_1, C_2, ..., C_n, so that relaxation is attempted on the best candidates first. The algorithm uses the set of partial orderings of features to help determine whether one candidate, C_i, is better than another, C_j. It works by generating a score for each feature ordering, F_j, and candidate, C_i. This score represents how well the speaker's description, D, relaxes to candidate C_i, while following the feature order F_j. The score is based on the number of features that have to be relaxed and how well the required relaxation fits the feature order. The lower the score, the better the feature order fits the candidate.

[33] This is true in the context of the water pump domain and the reference system I built. I do not, however, claim that this is true in general. In fact, the number of combinations grows exponentially with respect to the number of links in the graph.

For the worst case, when all features must be relaxed in D to fit C_i, the score is infinite since the denominator is zero. For the best case, when no features of D need to be relaxed to fit C_i, the score is zero since the numerator is zero. This scoring technique works reasonably well since the main goal is to notice the extremes (the best and worst) and not to distinguish precisely the instances of candidates and feature orderings that are almost as good as another. Once all the (C_i,F_j) scores are collected, an overall score is generated from them for each C_i. That score is used to order the candidates for relaxation. It is the sum of the previous scores for some candidate C_i, i.e., the sum of all the SCORE(C_i,F_j), where i is a constant, and j runs from 1 to n. The candidates with the lowest overall scores are the best candidates.

Figure 6-4 provides a graphic illustration of what the best candidate algorithm does. A set of objects in the real world are selected by the partial matcher as potential candidates for the referent. These candidates are shown across the top of the figure. The lines on the right side of each box correspond to the set of feature values that describe that object. The speaker's description is represented in the center of the figure. The set of specified features and their assigned feature value (e.g., the pair Color-Maroon) are also shown there. A set of partial orderings are generated that suggest which features in the speaker's description should be relaxed first - one ordering for each knowledge source (shown as "Linguistic," "Perceptual," and "Hierarchical" in the figure). These are put together to form a directed graph that represents the possible, reasonable ways to relax the features specified in the speaker's description. While loops are shown in the directed graph, the algorithm will not follow them since one relaxes a feature only once. In fact, this graph isn't actually built by the best candidate algorithm but helps illustrate here the consideration of all the partial orderings by the algorithm. Finally, the referent candidates are reordered using the information expressed in the speaker's description and in the directed graph of features.

6.4.3 Determine which relaxation methods to apply

Once a set of ordered, potential candidates is selected, the relaxation mechanism begins step 3 of relaxation; it tries to find proper relaxation methods to relax the features that have just been ordered (success in finding such methods "justifies"

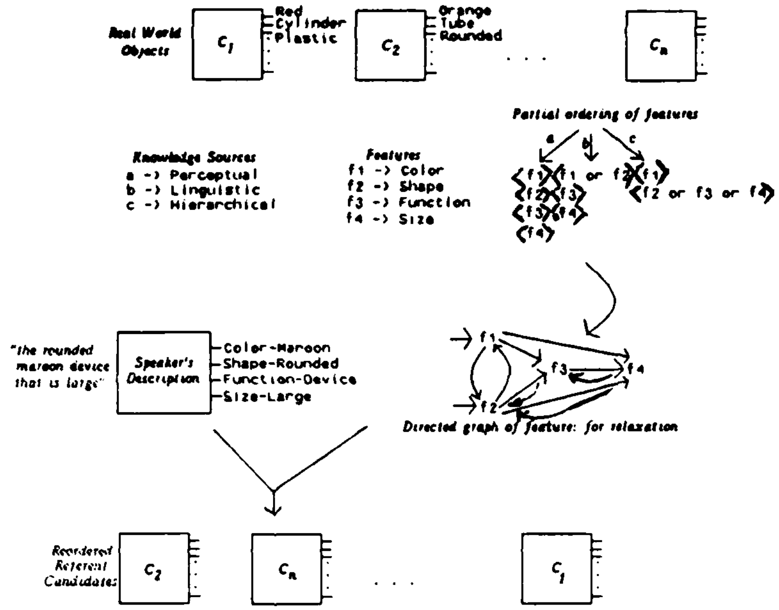


Figure 6-4: Reordering referent candidates

relaxing the speaker's description to the candidate). It stops at the first candidate in the list of candidates which falls below some threshold that is based on the strength of the relaxation methods that were used. For example, a relaxation method that relaxes "red" to "orange" is better than one that relaxes "red" to "blue." People perform similarly - once they find one referent that looks reasonable, they stop looking for others. If something goes wrong, such as an action failing, they simply retract their choice and try another. My algorithm permits the retraction of a selected candidate and the resumption of the testing of other candidates.

Relaxation can take place with many aspects of a speaker's description: with complex relations specified in the description, with individual features of a referent specified by the description, and with the focus of attention in the real world where one attempts to find a match. Complex relations specified in a speaker's description include spatial relations (e.g., "the outlet *near* the *top* of the tube"), comparatives (e.g., "the *larger* tube") and superlatives (e.g., "the *longest* tube"). These can be relaxed. The simpler features of an object (such as size or color) that are specified in the speaker's description are also open to relaxation.

Relaxation of a description has a few global strategies that can be followed for each part of the description: (1) drop the errorful feature value from the description altogether, (2) weaken or tighten the feature value in a principled way keeping its new value close to the specified one (e.g., movement within a subsumption hierarchy of feature values), or (3) try some other feature value based on some outside information (e.g., knowing that people often confuse opposite word pairs such as using "hole" for "peg" as illustrated in Excerpt 9).

Often the objects in focus in the real world implicitly cause other objects to be in focus [30, 78]. The subparts of an object in focus, for example, are reasonable candidates for the referent of a failing description and should be checked first before relaxing the description. At other times, the speaker might attribute features of a subpart of an object to the whole object (e.g., describing a plunger that is composed of a red handle, a metal rod, a blue cap, and a green cup as "the green plunger"). In these cases, the relaxation mechanism utilizes the part-whole relation in object descriptions to suggest a way to relax the speaker's description.

These strategies are realized through a set of procedures (or *relaxation methods*) that are organized hierarchically. Each procedure is an expert at relaxing its particular type of feature. For example, a Generate-Similar-Feature-Values procedure is composed of procedures like Generate-Similar-Shape-Values, Generate-Similar-Color-Values and Generate-Similar-Size-Values. Each of those procedures are specialists that attempt to first relax the feature value to one "near" or somehow "related" to the current one (e.g., one would prefer to first relax the color "red" to "pink" before relaxing it to "blue") and then, if that fails, to try relaxing it to any of the other possible values.[34] The effect of the latter case is really the same as if the feature was simply ignored.

For example, consider the relaxation procedure Generate-Similar-Shape-Values. It determines whether or not it is reasonable to relax one shape, the shape value specified in the speaker's description, to another shape, the one exhibited by the

[34] The latter case is there primarily for the times when one can't easily define a similarity metric for a feature. [47, 75] provide additional discussions about similarity metrics.

referent candidate. It determines how close the two shapes are by examining knowledge about shapes that is organized in a subsumption hierarchy. Figure 6-5 provides a piece of such a knowledge base. It shows that relaxing "hippopotamus-shaped" to the less specific shapes "elliptical" or "round" is reasonable. It also demonstrates that relaxing "hippopotamus-shaped" to "circular" is less reasonable. Similarly, relaxing "cylindrical" to "globe-shaped" isn't very reasonable.

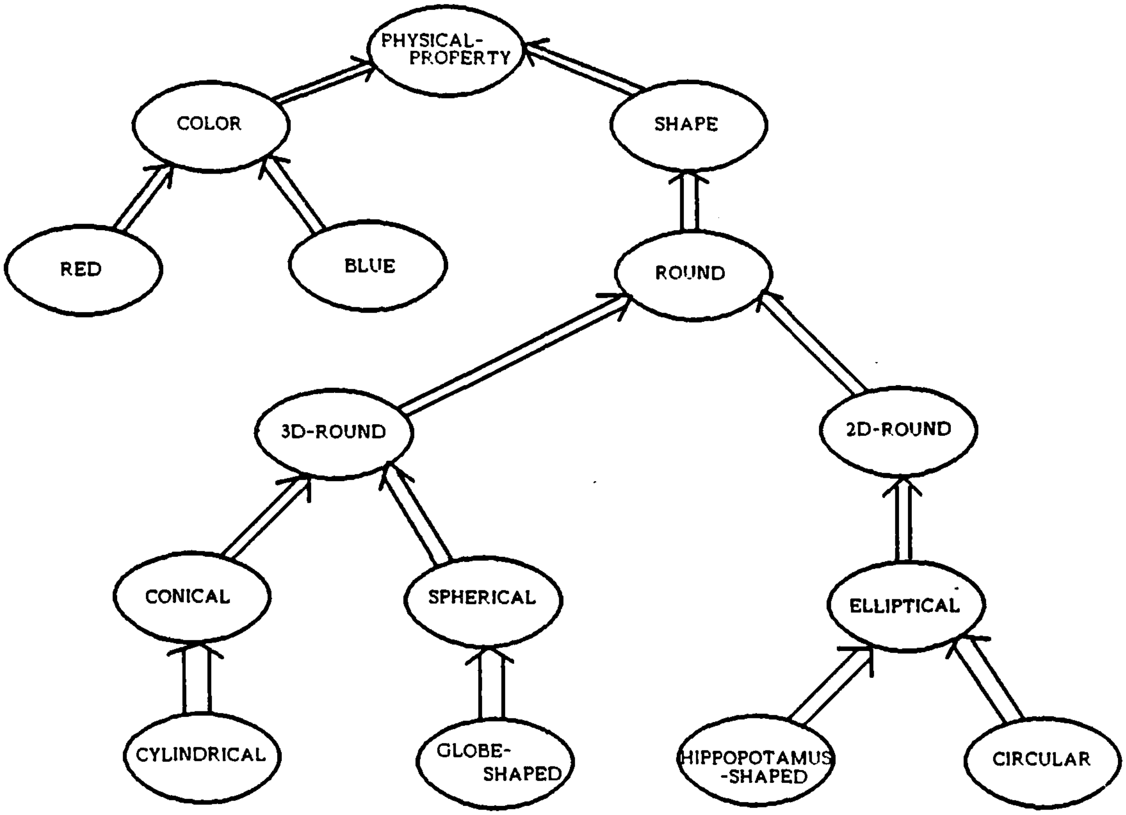


Figure 6-5: Sample hierarchy of feature values

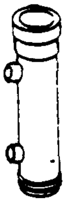
Not all forms of relaxation work well using a hierarchical knowledge base. The relaxation of color values is one such example. Most colors are viewed as reasonably distinct values so it would be hard to represent in a hierarchy the relationship between two colors. In cases like that for color, special routines must be used that know which feature values are related. I described in Chapter 5 a series of Similar-Color assertions whose purpose was to provide a general data base for such a routine. One color, Color1, can be relaxed to another, Color2, if Similar-Color(Color1,Color2) is true. Even if that predicate is false, Color1 could still be relaxed to Color2, though the relaxation is much less reasonable. If the system/listener knows that the speaker confuses certain colors from previous experience with the speaker, then those confusions can be represented as rules that can guide the relaxation. For example, if I know that the speaker often describes orange objects as red, then I would be more willing to substitute "orange" for "red" in the speaker's description.

6.5 An example of misreference resolution

This section describes how a referent identification system can recover from a misreference using the scheme outlined in the previous section. For the purposes of this example, assume that the water pump objects currently in focus include the *CAP*, the *MAINTUBE*, the *AIRCHAMBER* and the *STAND*. Assume also that the speaker tries to describe two of the objects – the *MAINTUBE* and the *AIRCHAMBER*.

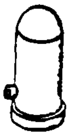
DescrA:
"...two devices that are clear plastic.

Stand


Main Tube

DescrB:
One of them has two openings on the outside with threads on the end, and its about five inches long.

Cap


Air Chamber

DescrC:
The other one is a rounded piece with a turquoise base on it.

DescrD:
Both are tubular.

DescrE:
The rounded piece fits loosely over..."

The reference system can find a unique referent for the first object (described by **DescrA**, **DescrB** and **DescrD**) but not for the second (described by **DescrA**, **DescrC**, **DescrD** and **DescrE**). The relaxation algorithm will be shown below to reduce the set of referent candidates for the second one down to two. It, then, requires the system/listener to try out those candidates to determine if one, or both, fits loosely. The protocols exhibit a similar result when the listener uses "fits loosely" to get the correct referent (e.g., Excerpt 6 exemplifies where "fit" is used by the speaker to help confirm that the proper referent was found). My system simulates this test by asking the user about the fit.

Figure 6-6 provides a simplified and linearized view of the actual KL-One representation of the speaker's descriptions after they have been parsed and semantically interpreted. A representation of each of the water pump objects that are currently under consideration (i.e., in focus) is presented in Figure 6-7. Each provides a physical description of the object - in terms of its dimensions, the basic 3-D shapes composing it, and its physical features - and a basic functional description of the object. The first entry in each representation in Figure 6-7 (that entry is shown in uppercase) defines the basic kind of entity being described (e.g., "TUBE" means that the object being described is some kind of tube). The words in mixed case refer to the names of features and the words in uppercase refer to possible fillers of those features from things in the water pump world. The "Subpart" feature provides a place for an embedded description of an object that is a subpart of a parent object. Such subparts can be referred to on their own or as part of the parent object. The "Orientation" feature, used in the representations in Figure 6-7, provides a rotation and translation of the object from some standard orientation to the object's current orientation in 3-D space. The standard orientation provides a way to define relative positions such as "top," "bottom," or "side." Figure 6-8 shows the KL-One taxonomy representing the same objects.

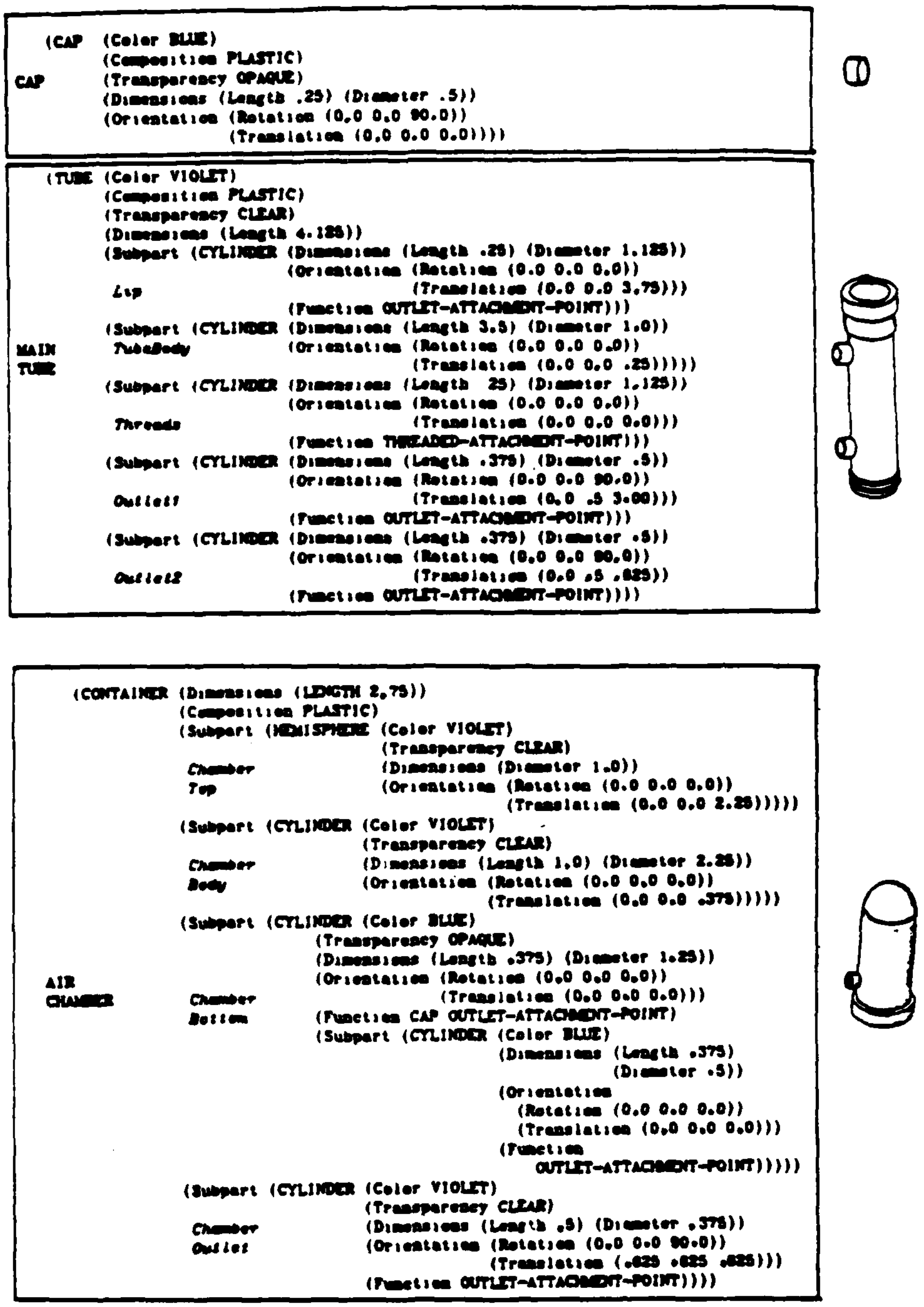
The first step in the reference process is the actual search for a referent in the knowledge base. In people, the reference identification process is incremental in nature, i.e., the listener can begin the search process before he hears the complete description. This was observed throughout the videotape excerpts. I try to simulate this incremental nature in my algorithm. It is readily apparent when considering the placement of the first description in **DescrD** into the KL-One taxonomy shown in Figure 6-8. **DescrD** is incrementally defined by first adding **DescrA** - as shown in Figure 6-9 - and then **DescrB** - as shown in Figure 6-11 - to the taxonomy. The KL-One Classifier compares the features specified in the speaker's descriptions with the features specified for each element in the KL-One taxonomy that corresponds to one of the current objects of interest in the real world. Notice that some features are directly comparable. For example, the "Transparency" feature of **DescrA** and the "Transparency" feature of *MAINTUBE* are both equal to "CLEAR." All the other

```
DescrA: (DEVICE (Transparency CLEAR)
                (Composition PLASTIC))
DescrB: (DEVICE (Transparency CLEAR)
                (Composition PLASTIC)
                (Subpart (OPENING))
                (Subpart (OPENING))
                (Subpart
                  (THREADS (Rel-Position END)))
                (Dimensions (Length 5.0)))
DescrC: (DEVICE (Transparency CLEAR)
                (Composition PLASTIC)
                (Shape ROUND)
                (Subpart (BASE (Color TURQUOISE))))
DescrD: (DEVICE (Transparency CLEAR)
                (Composition PLASTIC)
                (Subpart (OPENING))
                (Subpart (OPENING))
                (Subpart
                  (THREADS (Rel-Position END)))
                (Dimensions (LENGTH 5.0))
                (Analogical-Shape TUBULAR))
        (DEVICE (Transparency CLEAR)
                (Composition PLASTIC)
                (Shape ROUND)
                (Analogical-Shape TUBULAR)
                (Subpart (BASE (Color TURQUOISE))))
DescrE: (FIT-INTO
          (Outer (DEVICE (Transparency CLEAR)
                         (Composition PLASTIC)
                         (Shape ROUND)
                         (Analogical-Shape TUBULAR)
                         (Subpart
                           (BASE (Color TURQUOISE)))))
          (Inner . . .)
          (FitCondition LOOSE))
```

Figure 6-6: The speaker's descriptions

features specified in **DescrA** fit the *MAINTUBE* so the *MAINTUBE* can be described by **DescrA**. This is illustrated in Figure 6-10 where *MAINTUBE* is shown as a subconcept of **DescrA**. *STAND* also is shown as a subconcept of **DescrA**. *AIR CHAMBER* is shown as a possible subconcept (with the dotted arrow) because **DescrA** mismatches with it on one of its subparts.[35] Other features require in-depth processing - that is outside the capability of the KL-One classifier - before they can be compared. The OPENING value of "Subpart" in **DescrB** provides a good example of this. Consider comparing it

[35] I am stretching the definition of KL-One here with the dotted subsumption arrow. The point I want to make is that the *AIRCHAMBER* is similar to **DescrA** because their descriptions are almost exactly the same.



```
   (CAP  (Color BLUE)
         (Composition PLASTIC)
CAP      (Transparency OPAQUE)
         (Dimensions (Length .25) (Diameter .5))
         (Orientation (Rotation (0.0 0.0 90.0))
                      (Translation (0.0 0.0 0.0))))
```

```
   (TUBE (Color VIOLET)
         (Composition PLASTIC)
         (Transparency CLEAR)
         (Dimensions (Length 4.125))
         (Subpart (CYLINDER (Dimensions (Length .25) (Diameter 1.125))
                            (Orientation (Rotation (0.0 0.0 0.0))
          Lip                            (Translation (0.0 0.0 3.75)))
                            (Function OUTLET-ATTACHMENT-POINT)))
         (Subpart (CYLINDER (Dimensions (Length 3.5) (Diameter 1.0))
MAIN      TubeBody          (Orientation (Rotation (0.0 0.0 0.0))
TUBE                                     (Translation (0.0 0.0 .25)))))
         (Subpart (CYLINDER (Dimensions (Length  25) (Diameter 1.125))
                            (Orientation (Rotation (0.0 0.0 0.0))
          Threads                        (Translation (0.0 0.0 0.0)))
                            (Function THREADED-ATTACHMENT-POINT)))
         (Subpart (CYLINDER (Dimensions (Length .375) (Diameter .5))
                            (Orientation (Rotation (0.0 0.0 90.0))
          Outlet1                        (Translation (0.0 .5 3.00)))
                            (Function OUTLET-ATTACHMENT-POINT)))
         (Subpart (CYLINDER (Dimensions (Length .375) (Diameter .5))
                            (Orientation (Rotation (0.0 0.0 90.0))
          Outlet2                        (Translation (0.0 .5 .825))
                            (Function OUTLET-ATTACHMENT-POINT))))
```

```
   (CONTAINER (Dimensions (LENGTH 2,75))
              (Composition PLASTIC)
              (Subpart (HEMISPHERE (Color VIOLET)
                                   (Transparency CLEAR)
               Chamber             (Dimensions (Diameter 1.0))
               Top                 (Orientation (Rotation (0.0 0.0 0.0))
                                                (Translation (0.0 0.0 2.25)))))
              (Subpart (CYLINDER (Color VIOLET)
                                 (Transparency CLEAR)
               Chamber           (Dimensions (Length 1,0) (Diameter 2.25))
               Body              (Orientation (Rotation (0.0 0.0 0.0))
                                              (Translation (0.0 0.0 .375)))))
              (Subpart (CYLINDER (Color BLUE)
                             (Transparency OPAQUE)
                             (Dimensions (Length .375) (Diameter 1.25))
AIR                          (Orientation (Rotation (0.0 0.0 0.0))
CHAMBER        Chamber                    (Translation (0.0 0.0 0.0)))
               Bottom        (Function CAP OUTLET-ATTACHMENT-POINT)
                             (Subpart (CYLINDER (Color BLUE)
                                                (Dimensions (Length .375)
                                                            (Diameter .5))
                                                (Orientation
                                                  (Rotation (0.0 0.0 0.0))
                                                  (Translation (0.0 0.0 0.0)))
                                                (Function
                                                   OUTLET-ATTACHMENT-POINT)))))
              (Subpart (CYLINDER (Color VIOLET)
                                 (Transparency CLEAR)
               Chamber           (Dimensions (Length .5) (Diameter .375))
               Outlet            (Orientation (Rotation (0.0 0.0 90.0))
                                              (Translation (.625 .625 .625)))
                                 (Function OUTLET-ATTACHMENT-POINT))))
```

Figure 6-7: The objects in focus

```
        (TUBE (Dimensions (Length 2.75))
              (Composition PLASTIC)
              (Subpart (CYLINDER (Color BLUE)
                                 (Transparency CLEAR)
               Top               (Dimensions (Length 2.25) (Diameter .375))
                                 (Orientation (Rotation (0.0 0.0 0.0))
  STAND                                       (Translation (.5 0.0 .375)))
                                 (Function OUTLET-ATTACHMENT-POINT)))
              (Subpart (CYLINDER (Color BLUE)
                                 (Transparency CLEAR)
               Base              (Dimensions (Length .375) (Diameter 1.0))
                                 (Orientation (Rotation (0.0 0.0 0.0))
                                              (Translation (0.0 0.0 0.0)))
                                 (Function OUTLET-ATTACHMENT-POINT))))
```

Figure 6-7, Concluded

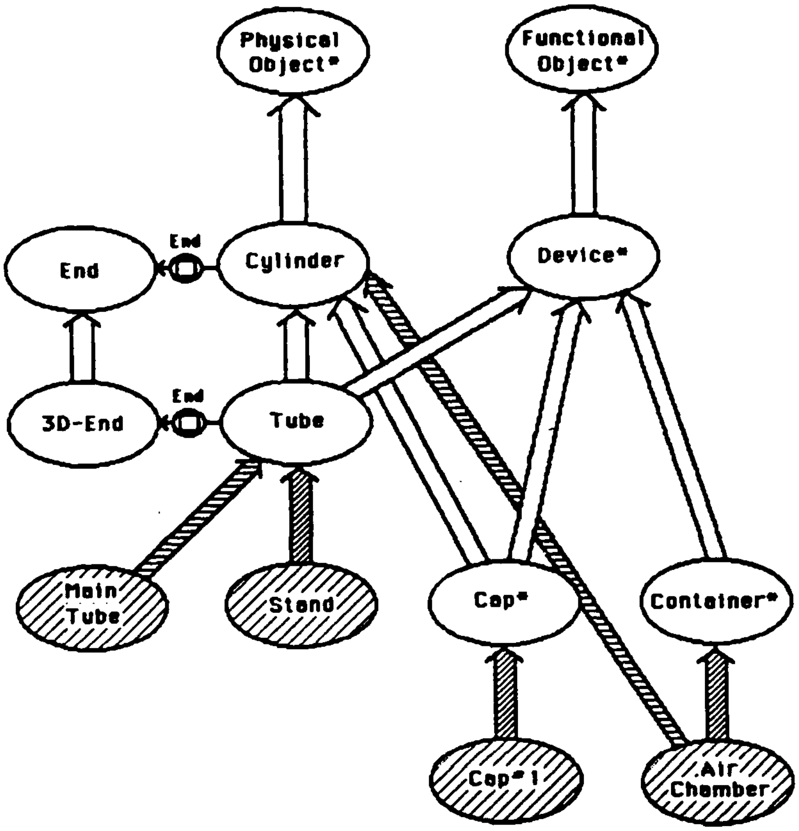


Figure 6-8: Taxonomy representing the objects in focus

to the "Subpart" entries for *MAINTUBE* shown in Figure 6-7. An OPENING, as seen in Figure 6-12, is thought of primarily as a 2-D cross-section (such as a "hole"), while the two CYLINDER subparts of *MAINTUBE* are viewed as (3-D) cylinders that have the "Function" of being outlets, i.e., OUTLET-ATTACHMENT-POINTS. To compare OPENING and one of the cylinders, say CYLINDER', the inference must be made that both things can describe the same thing (similar inferences are developed in [43]). One way this inference can occur is by recursively examining the subparts of *MAINTUBE* (and their subparts, etc.) with the KL-One partial matcher until the cylinders are examined at the 2-D level. At that level, an end of the cylinder will be defined as an OPENING. With that examination, the *MAINTUBE* can be seen as described by **DescrB**. This inference process is illustrated in Figure 6-12. There the partial matcher examines the roles Lip, Outlet1, and Outlet2 of *MAINTUBE* which represents its subparts and determines the following:

- A *CYLINDER* can have an *End* which is either a *2D-End* (e.g., a lid or hole) or a *3D-End* (e.g., a lip).
- A *2D-End* is either an *OPEN-2D-END* (e.g., a hole) or a *CLOSED-2D-END* (e.g., a lid on a can).
- An *OPEN-2D-END* is a kind of *OPEN-2D-OBJECT*.

These facts imply that *OPENING* can match any of the subparts Lip, Outlet1, or Outlet2 on *MAINTUBE* since those subparts are defined as cylinders that function as outlets (i.e., *Outlet-Attachment-Points*).

DescrC poses different problems. **DescrC** refers to an object that is supposed to have a subpart that is TURQUOISE. The Classifier determines that **DescrC** could not describe either the *CAP* or *STAND* because both are BLUE. It also could not describe the *MAINTUBE*[36] or *AIR CHAMBER* since each has subparts that are either VIOLET or BLUE. The Classifier places **DescrC** as best it can in the taxonomy, showing no connections between it and any of the objects currently in focus. **DescrD** provides no further help and is similarly placed. This is shown in Figure 6-13. At this point, a probable misreference is noted. The reference mechanism now tries to find potential referent candidates, using the taxonomy exploration routine described in Section 6.4.1, by examining the elements closest to **DescrD** in the taxonomy and using the partial

[36] Since **DescrB** refers to *MAINTUBE*, *MAINTUBE* could be dropped as a potential referent candidate for **DescrC**. I will, however, leave it as a potential candidate to make this example more complex.

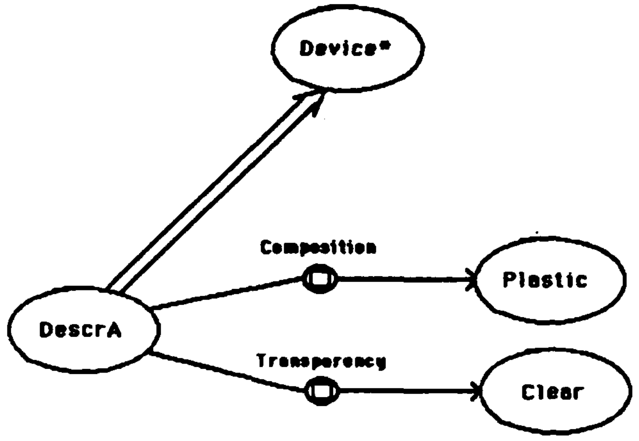


Figure 6-9: Adding DescrA to the taxonomy

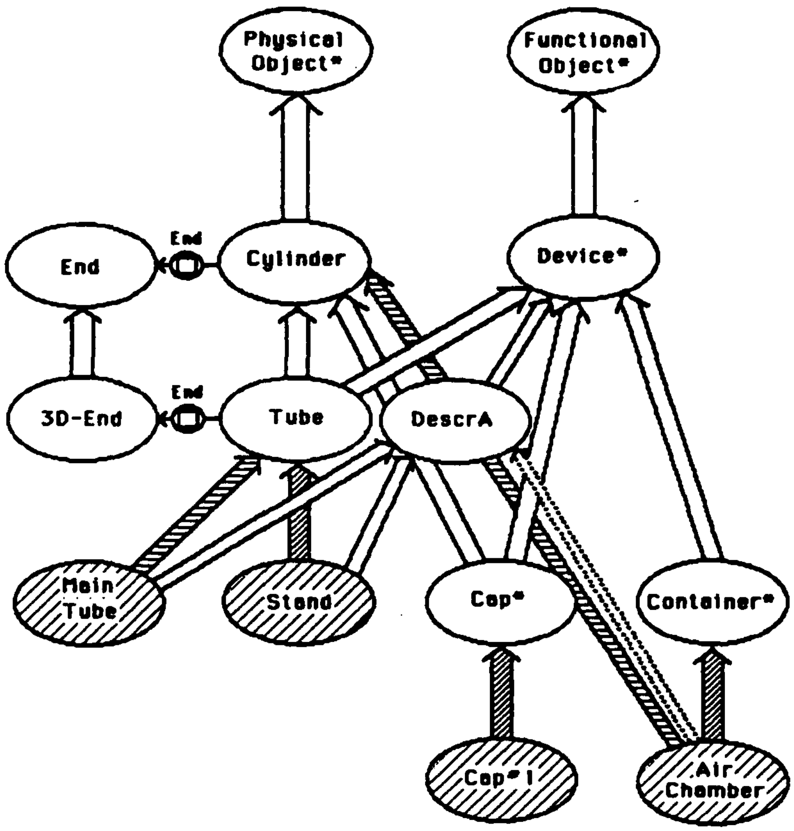


Figure 6-10: The classified DescrA

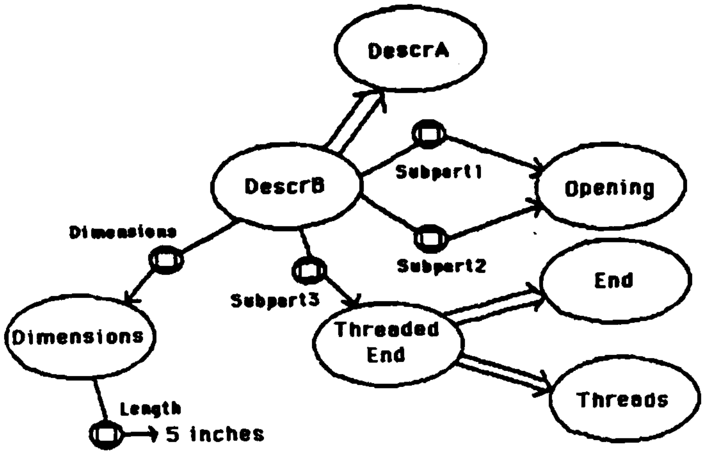


Figure 6-11: Adding DescrB to the taxonomy

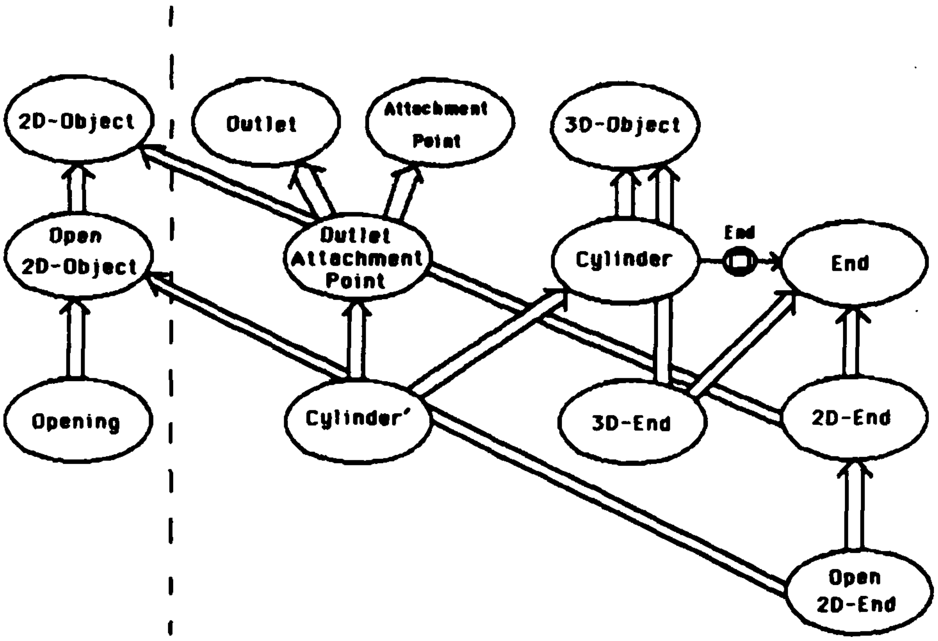


Figure 6-12: Attempt to match OPENING to CYLINDER'

matcher to score how close each element is to DescrD.[37] This is illustrated in Figure 6-14. The matcher determines *MAINTUBE*, *STAND*, and *AIR CHAMBER* as reasonable candidates by aligning and comparing their features to DescrD.

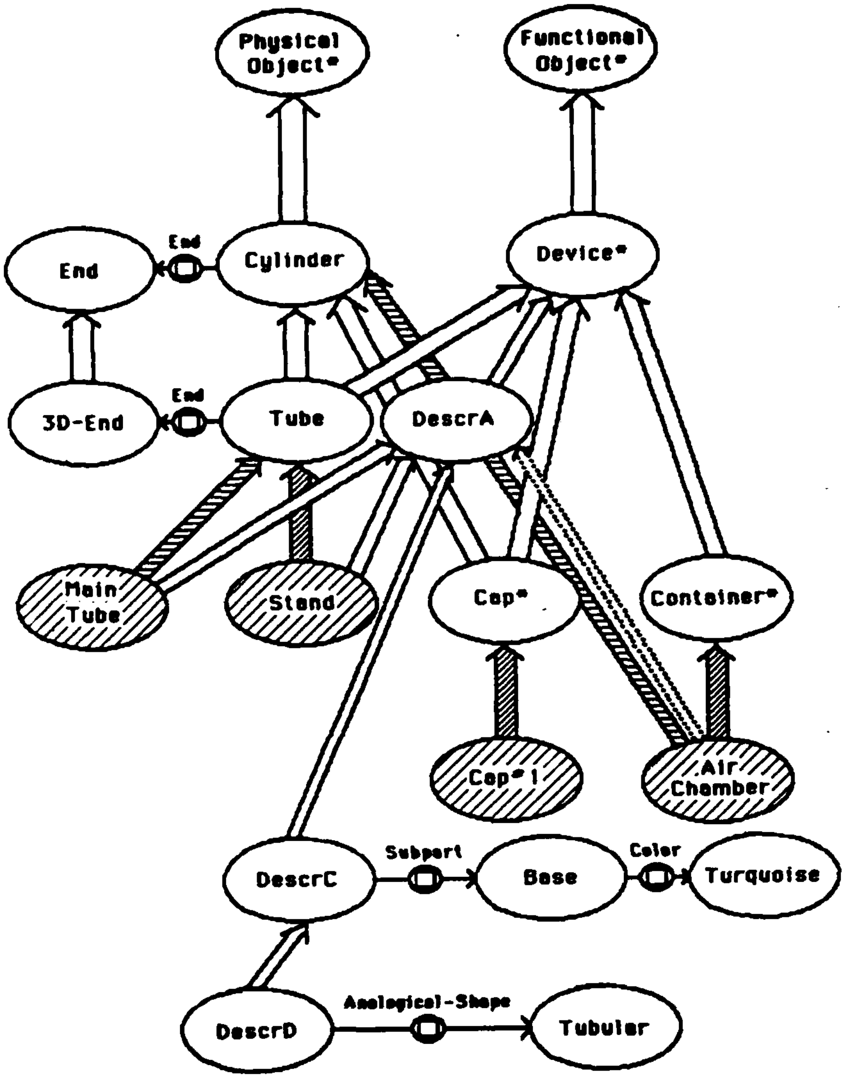


Figure 6-13: Adding DescrC and DescrD to the taxonomy

[37] The partial matcher scores are numerical scores computed from a set of role scores that indicate how well each feature of the two descriptions match. Those feature scores are represented on a scale: {+}, {> or <}, {=}, {?}, {-}. + is the highest and - is the lowest score. > and < have the same score but the algorithm can distinguish between them.

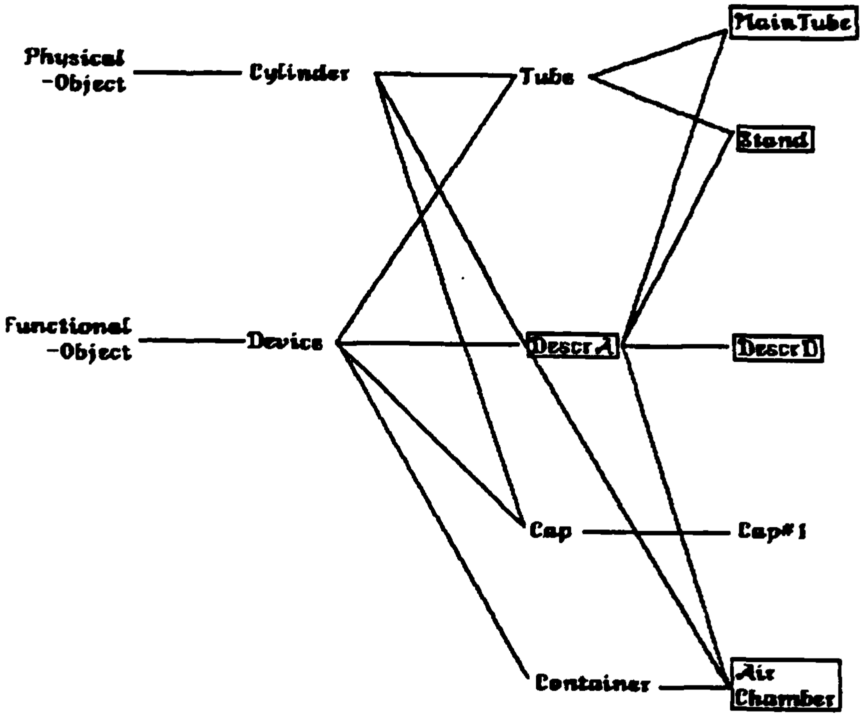


Figure 6-14: Exploring the taxonomy for referent candidates

Scoring **DescrD** to *MAINTUBE*:

- a TUBE is a kind of DEVICE; (>)
- the Transparency of each is CLEAR; (+)
- the Composition of each is PLASTIC; (+)
- a TUBE implies Analogical-Shape TUBULAR, which implies Shape CYLINDRICAL, which is a kind of Shape ROUND; (>)
- the recursive partial matching of subparts: A BASE is viewed as a kind of BOTTOM. Therefore, BASE in **DescrD** could match to the subpart in *MAINTUBE* that has a Translation of (0.0 0.0 0.0) - i.e., *Threads* of *MAINTUBE*. However, they mismatch since color TURQUOISE in **DescrD** differs from color VIOLET of *MAINTUBE*. (-)

Scoring **DescrD** to *STAND*:

- a TUBE is a kind of DEVICE; (>)
- the Transparency of each is CLEAR; (+)

- the Composition of each is PLASTIC; (+)
- a TUBE implies Analogical-Shape TUBULAR, which implies Shape CYLINDRICAL, which is a kind of Shape ROUND; (>)
- the recursive partial matching of subparts: BASE in **DescrD** could match to the subpart in *STAND* that has a Translation of (0.0 0.0 0.0) – i.e., *Base* of *STAND*. However, they mismatch since color TURQUOISE in **DescrD** differs from color BLUE of *STAND*. (–)

Scoring **DescrD** to *AIR CHAMBER*:

- a CONTAINER is a kind of DEVICE; (>)
- the Transparency of **DescrD**, CLEAR, matches the Transparency of *ChamberTop*, *ChamberOutlet* and *ChamberBody* of *AIR CHAMBER* but mismatches the Transparency of *ChamberBottom* of *AIR CHAMBER*. Therefore, the partial match is uncertain; (?)
- the Composition of each is PLASTIC; (+)
- the subparts of *AIR CHAMBER* have Shape HEMISPHERICAL and CYLINDRICAL which are each a kind of Shape ROUND; (>)
- the recursive partial matching of subparts: BASE in **DescrD** could match to the subpart in *AIR CHAMBER* that has a translation of (0.0 0.0 0.0) – i.e., *ChamberBottom* of *AIR CHAMBER*. However, they mismatch since color TURQUOISE in **DescrD** differs from color BLUE of *AIR CHAMBER*. (–)

Figure 6–15 summarizes the scoring. A weighted, overall numerical score is generated from the scores shown there.

The above analysis using the partial matcher provides no clear winner since the differences are so close causing the scores generated for the candidates to be almost exactly the same (i.e., the only difference was in the score for Transparency). All candidates, hence, will be retained for now

At this point, the knowledge sources and their associated rules that were mentioned earlier apply. These rules attempt to order the feature values in the speaker's description for relaxation. First, we'll order the features in **DescrD** using linguistic knowledge. Linguistic analysis of **DescrD**, " are clear plastic ... a rounded piece with a turquoise base ... Both are tubular fits loosely over ...," tells us that the features were specified using the following modifiers

DescrD

	SuperC	Composition	Transparency	Shape	Subparts
Maintube	>	+	+	>	-
Stand	>	+	+	>	-
Air Chamber	>	+	?	>	-

Range of role scores:

Low Correlation - ? = < > + High Correlation

Figure 6-15: Scoring DescrD to the referent candidates

- o Adjective: (Shape ROUND)
- o Prepositional Phrase: (Subpart (BASE (Color TURQUOISE)))
- o Predicate Complement: (Transparency CLEAR), (Composition PLASTIC), (Analogical-Shape TUBULAR), (Fit LOOSE)

Observations from the protocols (as described by the rules developed in Chapter 5) has shown that people tend to relax first those features specified as adjectives, then as prepositional phrases and finally as relative clauses or predicate complements. Figure 6-2 shows this rule. The rule suggests relaxation of **DescrD** in the order:

```
{Shape} < {Color,Subpart}
        < {Transparency,Composition,Analogical-Shape,Fit}.
```

The set of features on the left side of a "<" symbol is relaxed before the set on the right side. The order that the features inside the braces, "{...}", are relaxed is left unspecified (i.e., any order of relaxation is alright). Perceptual information about the domain also provides suggestions. Whenever a feature has feature values that are close, then one should be prepared to relax any of them to any of the others (I call this the "clustered feature value rule"). Figure 6-16 illustrates a set of assertions that compose a data base of similar color values in some domain. The Similar-Color predicate is defined to be reflexive and symmetric but not transitive. In this example, since a number of the color pairs are very close, color may be a reasonable thing to relax (see Figure 6-17). The clustered color rule defined in Figure 6-18 would suggest such a relaxation. It requires that there are at least three objects in the world that have similar colors. It is meant as an exemplar for a whole series of rules (e.g., ClusteredShapeValues, ClusteredTransparencyValues, and so on). Hierarchical information about how closely related one feature value is to another can also be used to determine what to relax. The Shape values are a good example as shown in Figure 6-19. A CYLINDRICAL shape is also a CONICAL shape, which is also a 3-D ROUND shape. Hence, it is very reasonable to match ROUNDED to CYLINDRICAL. All of these suggestions can be put together to form the order:

```
{Shape,Color} < {Subport}
          < {Transparency,Composition,
                    Analogical-Shape,Fit}.
```

```
Similar-Color ("BLUE","VIOLET")←
Similar-Color ("BLUE","TURQUOISE")←
Similar-Color ("GREEN","TURQUOISE")←
Similar-Color ("RED","PINK")←
Similar-Color ("RED","MAROON")←
Similar-Color ("RED","MAGENTA")←
      • • •
```

Figure 6-16: Similar color values

Colors of Candidates & DescrD	MainTube- violet Stand- blue Air Chamber- violet, blue DescrD- turquoise

Retrieve those Similar-Color assertions in the data base for the colors BLUE, VIOLET and TURQUOISE.

```
Similar-Color("BLUE","VIOLET")←
Similar-Color("BLUE","TURQUOISE")←
Similar-Color("GREEN","TURQUOISE")←

    ...
```

Figure 6-17: Objects with similar colors

The referent candidates *MAINTUBE*, *STAND*, and *AIR CHAMBER* can be examined

One can relax a feature whose feature values are clustered closely together before those of a non-clustered feature.

```
ClusteredFeatureValues(COLOR,w)
  ←Feature(COLOR),World(w),
    ColorValue(c1),ColorValue(c2),ColorValue(c3),
    WorldObj(o1,w),WorldObj(o2,w),WorldObj(o3,w),
    Color(c1,o1),Color(c2,o2),Color(c3,o3),
    Similar-Color(c1,c2),Similar-Color(c1,c3),
    Similar-Color(c2,c3)

 Relax-Feature-Before(v1,v2)
   ←ClusteredFeatureValues(feature(v1),w),
     NOT(ClusteredFeatureValues(feature(v2),w))
```

Figure 6–18: The clustered color value rule

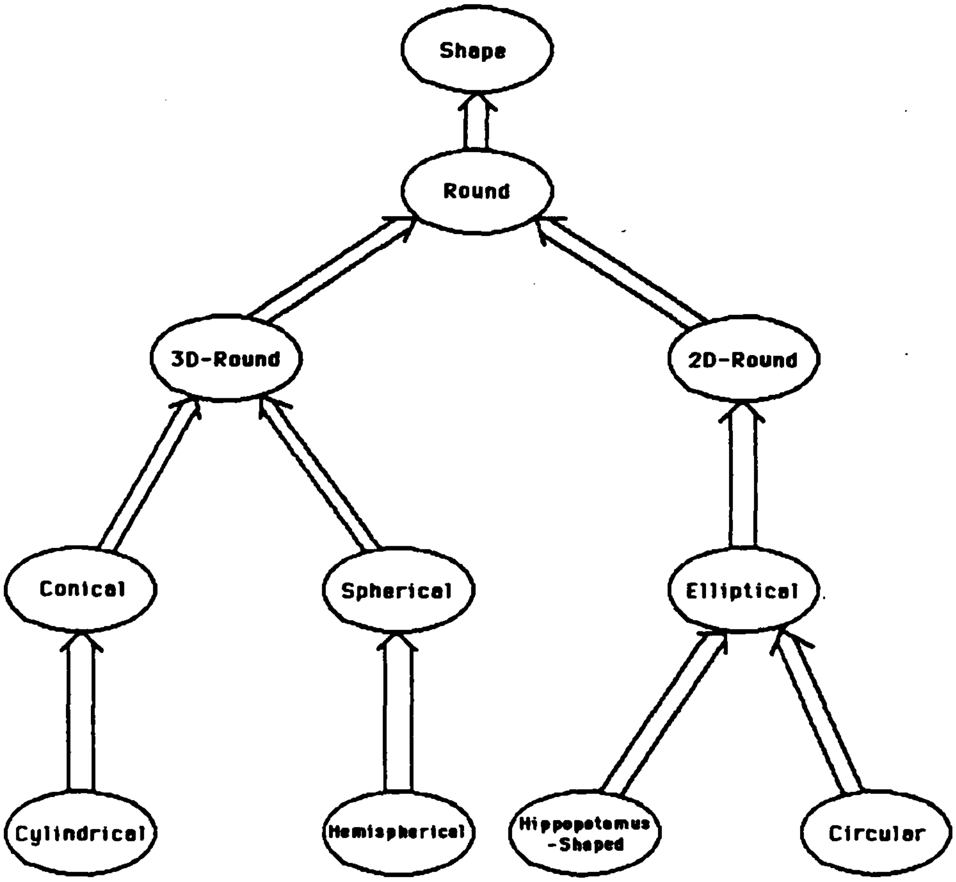


Figure 6–19: Hierarchical shape knowledge

and possibly ordered themselves using the above feature ordering. For this example, the relaxation of **DescrD** to any of the candidates requires relaxing their SHAPE and COLOR features. Since they each require relaxing the same features, the candidates can not be ordered with respect to each other (i.e., none of the possible feature orders is better for relaxing the candidates). Hence, no one candidate stands out as the most likely referent.

While no ordering of the candidates was possible, the order generated to relax the features in the speaker's description can still be used to guide the relaxation of each candidate. The relaxation methods mentioned at the end of the last section come into use here. Consider the shape values. The goal is to see if the ROUND shape specified in the speaker's description is similar to the shape values of each candidate. Generate-Similar-Shape-Values determines that it is reasonable to match ROUND to either the CYLINDRICAL or HEMISPHERICAL shapes of the *AIR CHAMBER* by examining the taxonomy shown in Figure 6-19 and noting that both shapes are below ROUND and 3D-ROUND. Notice that it is less reasonable to match CYLINDRICAL to HEMISPHERICAL since they are in different branches of the taxonomy. This holds equally true for the CYLINDRICAL shapes of the *MAINTUBE* and the *STAND*. Generate-Similar-Color-Values next tries relaxing the Color TURQUOISE. The assertions Similar-Color("BLUE","TURQUOISE")<- and Similar-Color("GREEN","TURQUOISE")<- are found as rules containing TURQUOISE. The colors BLUE and GREEN are, thus, the best alternates. Here only two clear winners exist - the *AIR CHAMBER* and the *STAND* - while the *MAINTUBE* is dropped as a candidate since it is reasonable to relax TURQUOISE to BLUE or to GREEN but not to VIOLET. Subpart, Transparency, Analogical-Shape, and Composition provide no further help (though, the fact that the *AIR CHAMBER* has both CLEAR and OPAQUE subparts could be used to put it slightly lower than the *STAND* whose subparts are all CLEAR. This difference, however, is not significant.). This leaves trial and error attempts to try to complete the FIT action specified in **DescrE**. The one (if any) that fits - and fits loosely - is selected as the referent. The protocols showed that people often do just that - reducing their set of choices down as best they can and then taking each of the remaining choices and trying out the requested action on them.

6.6 Summary

This chapter described the relaxation component of my reference identification mechanism. It divided the component into numerous subcomponents: a routine that explored for candidates, a partial matcher that scored how close those candidates were to the speaker's description, an ordering scheme that proposed what order to relax features in the speaker's description, a control structure that enforces that order, and relaxation methods that guided the actual relaxation of the speaker's description to fit a reasonable referent candidate in the world.

7. CONCLUSION AND FUTURE DIRECTIONS

This chapter summarizes the goals and accomplishments of this work and points out directions for future research.

7.1 The goals and accomplishments

My goal in this thesis was to build robust natural language understanding systems, allowing them to detect and avoid miscommunication. The goal was not to make a perfect listener but a more tolerant one that could avoid many mistakes, though it may still be wrong on occasion. In Chapter 2, I introduced a taxonomy of miscommunication problems that occur in expert-apprentice dialogues. I showed that reference mistakes are one kind of obstacle to robust communication. To tackle reference errors, I described how to extend the succeed/fail paradigm followed by previous natural language researchers. I developed a new way to look at reference that involves a more active, introspective approach to repairing communication.

I represented real world objects hierarchically in a knowledge base using a representation language, KL-One, that follows in the tradition of semantic networks and frames. In such a representation framework, the reference identification task looks for a referent by comparing the representation of the speaker's input to elements in the knowledge base by using a matching procedure. Failure to find a referent in previous reference identification systems resulted in the unsuccessful termination of the reference task. I claimed that people behave better than this and explicitly illustrated such cases in an expert-apprentice domain about toy water pumps.

I developed a theory of relaxation for recovering from reference failures that provides a much better model for human performance. When people are asked to identify objects, they behave in a particular way: find candidates, adjust as necessary, re-try, and, if necessary, give up and ask for help. I claim that relaxation is an integral part of this process and that the particular parameters of relaxation differ from task to task and person to person. My work models the relaxation process and provides a computational model for experimenting with the different parameters.

The theory incorporates the same language and physical knowledge that people use in performing reference identification to guide the relaxation process. This knowledge is represented as a set of rules and as data in a hierarchical knowledge base. Rule-based relaxation provided a methodical way to use knowledge about language and the world to find a referent. The hierarchical representation made it possible to tackle issues of imprecision and over-specification in a speaker's description. It allowed one to check the position of a description in the hierarchy and to use that position to judge imprecision and over-specification and to suggest possible repairs to the description.

Interestingly, one would expect that "closest" match would suffice to solve the problem of finding a referent. I showed, however, that it doesn't usually provide you with the correct referent. Closest match isn't sufficient because there are many features associated with an object and, thus, determining which of those features to keep and which to drop is a difficult problem due to the combinatorics and the effects of context. The relaxation method described circumvents the problem by using the knowledge that people have about language and the physical world to prune down the search space.

7.2 Future directions

There are many issues related to reference identification and recovery from reference failure that I did not address in either my theory or implementation. There are also other kinds of miscommunication beyond those due to reference that I described in Chapter 2 but that my theory does not attempt to deal with. I describe below some of the problems with my current reference system and then propose future research to handle them and to handle other types of miscommunication.

7.2.1 Deficiencies in the FWIM model

My current FWIM reference model has some immediate problems. First, I don't define when FWIM should simply give up and fail. I stated that whenever the relaxation of the speaker's description to a referent candidate falls below a certain "threshold," that relaxation should not take place. I failed, however, to define just how such a threshold is measured. Currently I view it as a percentage of the features relaxed in the speaker's description to the total number of features in the description. This suffices in many cases but neglects the particular relaxations that occur for each feature value in the speaker's description. If one of the particular feature value relaxations is unusual (e.g., relaxing "pink" to "black"), then it may be better for the whole relaxation of the description to fail. Any global threshold, thus, should take into account the goodness of the local feature value relaxations. Second, there are many kinds of references that I haven't considered. One of these is metonymical references. Metonomy occurs when one uses the name of one thing for that of another of which it is an attribute or with which it is associated. Consider the three descriptions below.[38] Notice how the noun phrase "the window" refers to three different things in each of the utterances.

"The window was broken." (the glass)
"The window was boarded up." (the opening)
"Open the window." (the glass/frame inset)

Third, my model has deficiencies as a cognitive model of how people perform reference. The processing is not as serial as I make it seem. There is a lot of competition among the different possibilities during the reference task and the effect of parallel interactions is probably much more complex than I have shown. For example, it seems a person probably does not consider one referent candidate at a time but might be considering several at once. Those kinds of interactions would probably add new and more complex kinds of support and hypotheses to those I already use.

7.2.2 Clarification dialogues and miscommunication

I ignored in my work the fact that a lot of miscommunication recovery is interactive, with the listener and speaker working together to correct the misunderstanding. Responding effectively to miscommunication often requires clarification dialogues (e.g., [41, 42]) to clarify the source of the error as well as to elucidate the speaker's goal. The point of clarification dialogues is not only to indicate that an error has occurred but to exchange enough information to pin down

[38] These examples were suggested to me by Bonnie Webber.

exactly what the misunderstanding is and how to fix it. Clarification dialogues are also a good place to catalogue errors – especially speaker dependent ones – so that they can be avoided in the future. For future work, I propose looking at how clarification dialogues are used in expert–apprentice tasks. They should be incorporated into my reference identification system and used whenever the system is unsure about a particular aspect of the relaxation process, or when the reference system is unable to relax the user's description.

While clarification dialogues provide a way to recover from miscommunication, they also have the potential of causing their own misunderstandings. Any attempt to use them to recover from miscommunication requires taking into account such potential problems. The next section describes some of the problems that can occur during clarifying dialogues and reasons for them.

7.2.2.1 Listener's expectations when things go wrong

Many speakers do a poor job recovering from errors that occur during a conversation. Part of the recovery problem is inherent in the modality of communication. The narrower the communication channel, the harder it is to detect the precise nature of an error and to explain to the listener what has gone wrong and how to rectify the situation. In general, however, the listener would expect the speaker to review the situation leading up to the mistake (negotiating with the listener along the way to discover what is wrong), to correct the mistake, and then continue cautiously from there.

When the speaker is notified of or detects confusion, one would expect him to (1) back up describing how the current state (i.e., the state of assembly in a task–oriented dialogue) should look and not just the current focus (unless possibly to determine if a mistake really has occurred by seeing if some situation holds), and then (2) move down in focus to items thought to be the cause of the confusion.[39] This is accomplished by decomposing the task into simpler pieces. In other words, when mistakes occur, a clarification sequence involves stepping back and describing how the

[39] Grosz [31, 32] describes this situation for failed descriptions. She states the expert anchors the description on some past action of the apprentice and then describes the object functionally.

object should currently look as a whole (and not describing just the elements currently in focus), and then moving down in focus to the elements thought to be the problem or to probe for them by trying to get the listener to identify any discrepancies. Once the mistake is discovered, the speaker explains how to undo it and then how to correctly proceed.

After a clarification sequence has been completed and the mistake corrected, the listener expects the speaker to "downshift" [20] in his descriptions. "Downshifting" entails being clearer from then on by going slower, putting less into each step and adding more descriptions – i.e., taking care to be specific <u>without</u> over-specifying. Once the listener demonstrates success again, the speaker will slowly "upshift" adding more complexity (and possibly less information) in each step.

7.2.3 Beyond reference miscommunication

Referring to things is just one of the ways people could miscommunicate. I primarily investigated reference problems but would like to explore other areas of miscommunication – especially ones where the discourse, speaker's intention and the requested actions are considered.

7.2.3.1 Subproblems of analyzing miscommunication

One of the first problems in analyzing miscommunication is the recognition of the kind of miscommunication occurring. We might have a case of reference failure, failure to understand the intentions of the speaker, failure of contextual disambiguation, or failure to interpret an imprecise, high-level request. The taxonomy of types of misunderstanding presented earlier in this thesis is a first step towards being able to recognize what kind of problem has occurred. I still need to explore ways of *recognizing each particular kind of problem*.

Another difficult problem is the development of rules for judging when more processing (e.g., determining if we need to request more information) is needed to understand a sequence of utterances. This requires specifying criteria for deciding whether the communicative goal is clearly understood, i.e., understanding when we have enough information to respond appropriately. One potential source of difficulty is utterances that are imprecise. This is primarily a problem for the plan recognition section of the BBN natural language system [19, 68].[40]

Imprecise utterances can cause trouble in two major ways: during the identification of their referents and while attempting to discover what goals are being specified by the speaker in the utterance. When a description of a referent is not complete enough to disambiguate a unique item among possible contenders, then the description is judged imprecise. A description is also considered imprecise when it provides so little information that a search of the available contenders is not possible because they could not be identified. Imprecision also occurs during goal and plan recognition (i.e., when the listener is trying to determine what the speaker is requesting him to do) when the goal is not clearly understood (e.g., when the literal meaning of an utterance differs from its intended goal), when not enough information is available to select a particular plan for carrying out the requested action, or when a selected plan has missing attributes that must be filled before the plan can be executed by the listener.

In addition to imprecise requests, I must *also* consider the *problem of ill-formed* utterances. These occur when the goal is confused (e.g., requesting something that does not fit with what one is trying to accomplish), when a request goes outside the capabilities of the system (here might consult a model of the system's capabilities), or when a particular goal or plan is clearly indicated but some aspects of it are not appropriately defined (e.g., an attribute is "filled" with an inappropriate value or some required information is not yet specified). It is important to be able to tell the difference between imprecise and ill-formed utterances.

7.2.3.2 Getting a handle on miscommunication

Incremental planning is a new planning model that allows for action descriptions and goal specifications to be handled in an incomplete way - they don't have to be perfectly specified. Currently proposed ideas by Vilain (see [73]) are to allow flexible and incremental handling along human lines. Incremental planning will allow a plan to be recognized over several utterances as more information comes in. It also will

[40] Also see [42] for a detailed integration of discourse analysis into plan recognition.

provide a way to re-plan should a flaw surface in the original plan. Incremental planning can give us a handle on the problems of coping with inappropriate utterances. It provides a place to consider multiple interpretations of the same utterance (e.g., when an utterance fails, the listener could reinterpret the utterance under a different plan or plan segment). Repair to the utterance could be directed by the incremental planner which might determine a more appropriate plan, might revise its earlier expectations (i.e., which plans seemed most relevant at this point in the conversation), or it might initiate a clarifying dialogue to determine the source of the problem.

However, one, needs more than the incremental planner to *recover from* miscommunication: (1) it is not capable of taking into account language level information that might influence recovery from a mistake (this need was clearly illustrated in Chapter 5 for reference failures), (2) it may be unable to be specific when it asks the speaker for clarification, and (3) it won't learn from the mistakes and adjust to a particular user's preferences. I propose another component outside the planner that tries to learn from mistakes by representing a generalization of the source of the mistake and the solution to it. It uses information found in the actual language of the speaker and information from the plan recognizer and the incremental planner. It consults the speaker when necessary by initiating a clarification dialogue.

APPENDIX A
REPRESENTING ACTIONS

Actions can also be represented in KL-One in a manner analogous to representing objects. Figure A-1 defines the action "PUT-INTO." The roles on concept PUT-INTO follow standard case analysis of verbs.

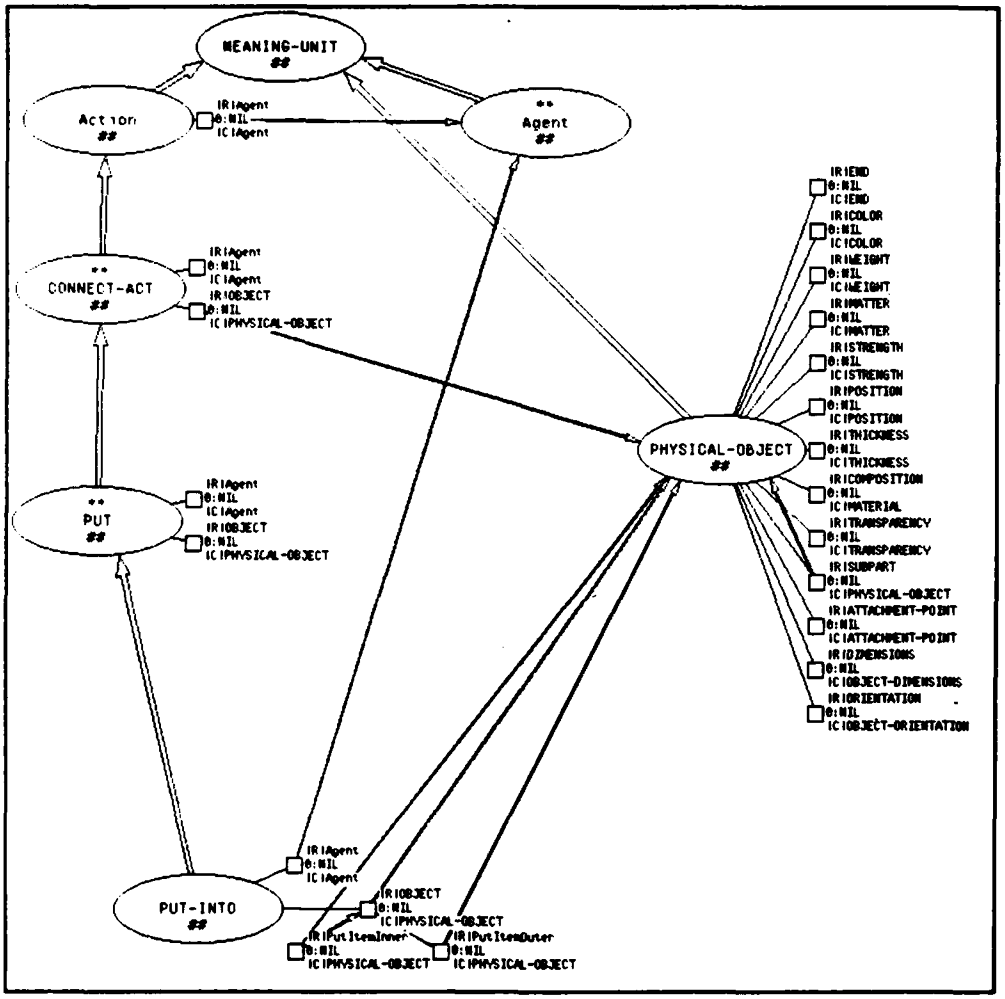


Figure A-1: Representation of the action "PUT-INTO"

APPENDIX B
PARSING AND SEMANTIC INTERPRETATION

My reference identification implementation was built by performing by hand all the parsing and semantic interpretation of test examples. To demonstrate how easy it would be to plug in a parser and semantic interpreter, I show in Figure B-1 an actual parse tree and semantic interpretation of the utterance "Get the large violet tube with two cylindrical outlets." They were generated using the IRUS parser and semantic interpreter [72]. The interpretation is in MRL, as described in Chapter 3. A sample interpretation rule used by the semantic interpreter is shown in Figure B-2.

```
PARSE (GET THE LARGE VIOLET TUBE WITH TWO CYLINDRICAL OUTLETS)


Interpretation:

(FOR THE TUBE40 / SUB-PART : (AND (COLOR TUBE40 VIOLET)
                                  (SIZE TUBE40 LARGE)
                                  (FUNCTION TUBE40 TUBE))
     ;
     (FOR 2 OUTLET41 / SUB-PART : (AND (SHAPE OUTLET41 CYLINDRICAL)
                                       (FUNCTION OUTLET41 OUTLET))
          ;
          (AND (PART-OF TUBE40 OUTLET41)
               (PICK-UP TUBE40))))


Parse Tree:

[IMPERATIVE#1
  OBJECT =
    [NP#2
      DET =
        {ART...THE...}
      PP =
        [[PP#3
           POBJ =
             [NP#4
               PARTITIVE =
                 [DETERMINER#5
                   QUANTITY =
                     [QUANTITY#6
                       NUMBER = 2]]
               ADJ =
                 [[ADJ#7
                    HEAD = CYLINDRICAL]]
               HEAD ={NOUN...OUTLETS...}]
           HEAD = WITH]]
      ADJ =
        [[ADJ#8
           HEAD = VIOLET]
         [ADJ#9
           HEAD = LARGE]]
      ADJ =
        [ADJ#8]
      HEAD ={NOUN...TUBE...}]
  HEAD = GET]
```

Figure B-1: A sample parse and semantic interpretation

```
Interpretation Rule:

     IRULE SUB-PART

           (NP HEAD * ADJ ((PROPERTY COLOR)
                (PROPERTY SIZE)
                (PROPERTY COMPLEXITY)
                (PROPERTY SHAPE))
               NOUN
               ((PROPERTY MATERIAL))
               PP
               ((PP HEAD WITH POBJ (SUPERC SUB-PART))
                (PP HEAD WITHOUT POBJ (SUPERC SUB-PART))
                (PP HEAD OF POBJ (SUPERC SUB-PART))
                (PP HEAD (PROPERTY LOCATION-PREP)
                    POBJ
                    (SUPERC PHYSICAL-OBJECT))))
           ==>
           (LIFTQUANTS ((PIECE (HEAD HEAD))
                        (PIECE-TYPE (PROPERTY HEAD SUB-PART))
                        (PIECE-COLOR (OPTIONAL (ADJ 1 HEAD)))
                        (PIECE-SIZE (OPTIONAL (ADJ 2 HEAD)))
                        (PIECE-COMPLEXITY (OPTIONAL (ADJ 3 HEAD)))
                        (PIECE-SHAPE (OPTIONAL (ADJ 4 HEAD)))
                        (PIECE-MATERIAL (OPTIONAL (NOUN 1 HEAD)))
                        (PIECE-1 (OPTIONAL (PP 1 POBJ)))
                        (PIECE-2 (OPTIONAL (PP 2 POBJ)))
                        (PIECE-3 (OPTIONAL (PP 3 POBJ)))
                        (LOCATION-PRED (OPTIONAL (PP 4 HEAD)))
                        (LOCATION (OPTIONAL (PP 4 POBJ))))
                       (CLASS SUB-PART)
                       (ANDPREDICATE (QUOTE (COLOR *V* PIECE-COLOR)))
                       (ANDPREDICATE (QUOTE (SIZE *V* PIECE-SIZE)))
                       (ANDPREDICATE (QUOTE (COMPLEXITY *V* PIECE-COMPLEXITY)))
                       (ANDPREDICATE (QUOTE (SHAPE *V* PIECE-SHAPE)))
                       (ANDPREDICATE (QUOTE (PIECE-TYPE *V* PIECE)))
                       (ANDPREDICATE (QUOTE (COMPOSITION *V* PIECE-MATERIAL)))
                       (ANDPREDICATE (QUOTE (PART-OF *V* PIECE-1)))
                       (ANDPREDICATE (QUOTE (NOT (PART-OF *V* PIECE-2))))
                       (ANDPREDICATE (QUOTE (PART-OF PIECE-3 *V*)))
                       (ANDPREDICATE (QUOTE (LOCATION-OF *V*
                                                (LOCATION-PRED LOCATION))))
                       )]
```

Figure B-2: A sample semantic interpretation rule

APPENDIX C
USING THE FOCUS MECHANISM

A focus mechanism has been written to simulate the shifting of focus between elements in both the dialogue and their correspondents in the real world. The current mechanism does not detect focus shifts but allows the user to manually intervene, through a series of pop-up menus, to force a focus shift. A more automated detection of focus shifts will be possible when the discourse tracker (or "focus machine" [69]) is added to the system.

Focus is used in my system to provide two related sets of partitions. One group of partitions divides up the KL-One representation of the linguistic world while the other group separates parts of the KL-One real world representation. The linguistic world categorizes the real world in the terms that people most often talk about it. The real world is meant to "model" a physical environment as it might be seen by a vision system. For example, a cylinder that has an extensional representative in the real world has a definite set of dimensions - its *length* and *diameter*. People, however, could describe the cylinder using less precise terms such as relative sizes like "big," "large," and "long," so such terms are part of the linguistic world. The real world is composed of basic 3-D shapes (e.g., generalized cylinders) while the linguistic world allows one to describe an object using analogical shapes (e.g., "the L-shaped tube"). The linguistic world is used to hold semantic interpretations of a speaker's utterances while the real world contains descriptions of the current physical world in front of the listener. Figure C-1 shows an example of a tube represented in the linguistic world and Figure C-2 shows a possible correspondent for it in the real world.

The focus mechanism interacts with the user to assign partitions of the real world. The user is presented a menu containing a list of objects in the world. There are several actions that can be performed on subsets of those objects. The user can create a new focus space[41], add elements to a previous focus space, or remove

[41] The term "focus space" here does not correspond exactly to the definition by Grosz [30] but is closer to her definition of "global focus." Here it is meant to encompass the current set (or "partition") of relevant objects. The most relevant object (i.e., the one currently under discussion) is considered to be the current "focus."

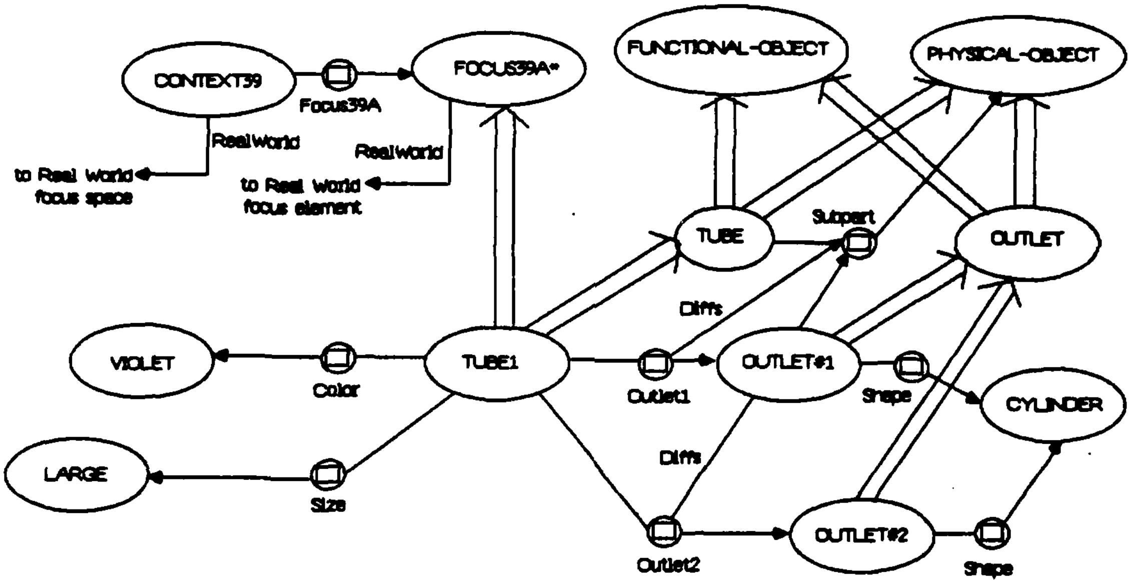


Figure C-1: A linguistic world tube

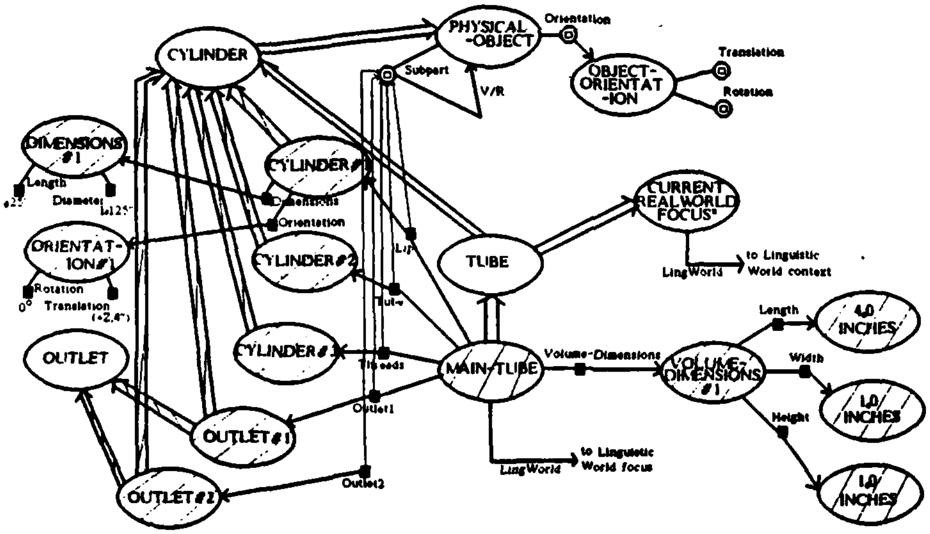


Figure C-2: A real world tube

elements from a previous focus space. Once an initial set of elements is assigned to a focus space, the focus system is ready to accept either an input, which is a semantic interpretation of a noun phrase constructed by RUS and PSI-KLONE, or a command to shift focus. This mechanism corresponds to the one actually used in understanding conversations, where listeners notice special discourse markers or recognize a change in the speaker's plan.

The semantic interpretation of a new input is placed into the linguistic world partition that is currently in focus.[42] This focus space has a pointer to the correspondent real world partition that describes the currently relevant real world elements and another pointer to the particular element that is the current focus of attention.[43] The new input automatically inherits the pointer to the real world element that is currently in focus. Unless there is some major discrepancy between the input and the real world focus, the referent of the input is assumed to be that real world element. Any discrepancies hint that a possible shift in focus has occurred or that the speaker has made a mistake. Another hint at a focus shift occurs if the current input contains information that was not previously mentioned. For example, if all previous inputs never mentioned the color of the object and now it is referred to as "the red thing," then the speaker may be hinting that focus should shift to something else [30].

When a shift of focus is indicated, the system pops up a menu asking where focus has shifted. The shift can be to a subpart of the object currently in focus, to another object, or to a subassembly that some previous action has built. Depending on which option is selected, another menu is generated that displays (1) a list of the subparts on the object currently in focus, (2) a list of the other objects, or (3) a list of subassemblies. The user selects the pa ticular subpart, object, or subassembly to which focus has shifted. At this point, the user must also say which focus space is to contain the new focus element. The focus space could be a new one (i.e., some new context involving the element) or a previous one where some action was left unresolved. If it is a previous one, it is identified to the user by showing him the set

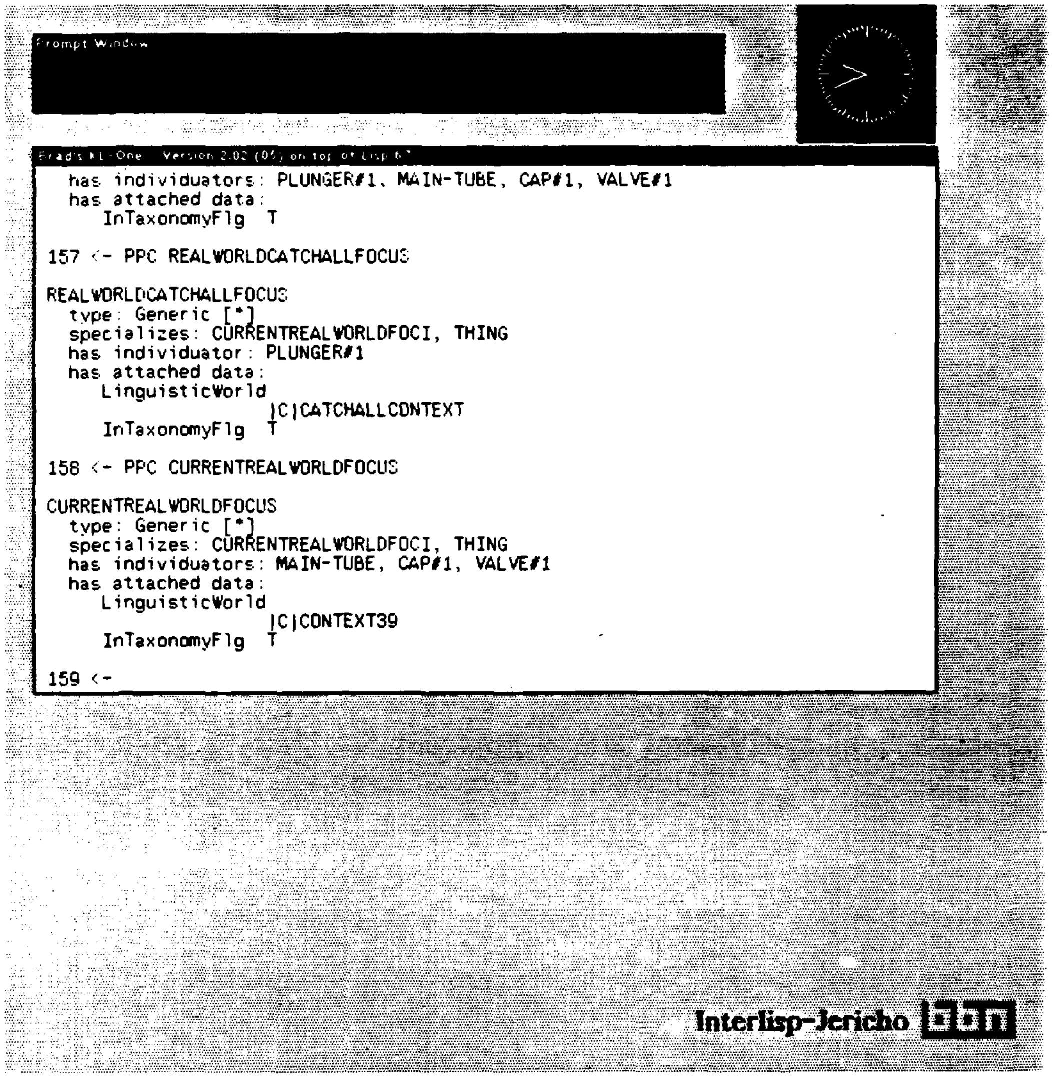
[42] This is achieved by adding a SuperC cable between the KL-One representation of the user's input and the KL-One representation of the current focus space.

[43] At the beginning of the dialogue, all elements in the real world are considered relevant but *no one element is the focus of attention.*

of objects already in that space. If it is a new one, the user selects which objects to place in that space.

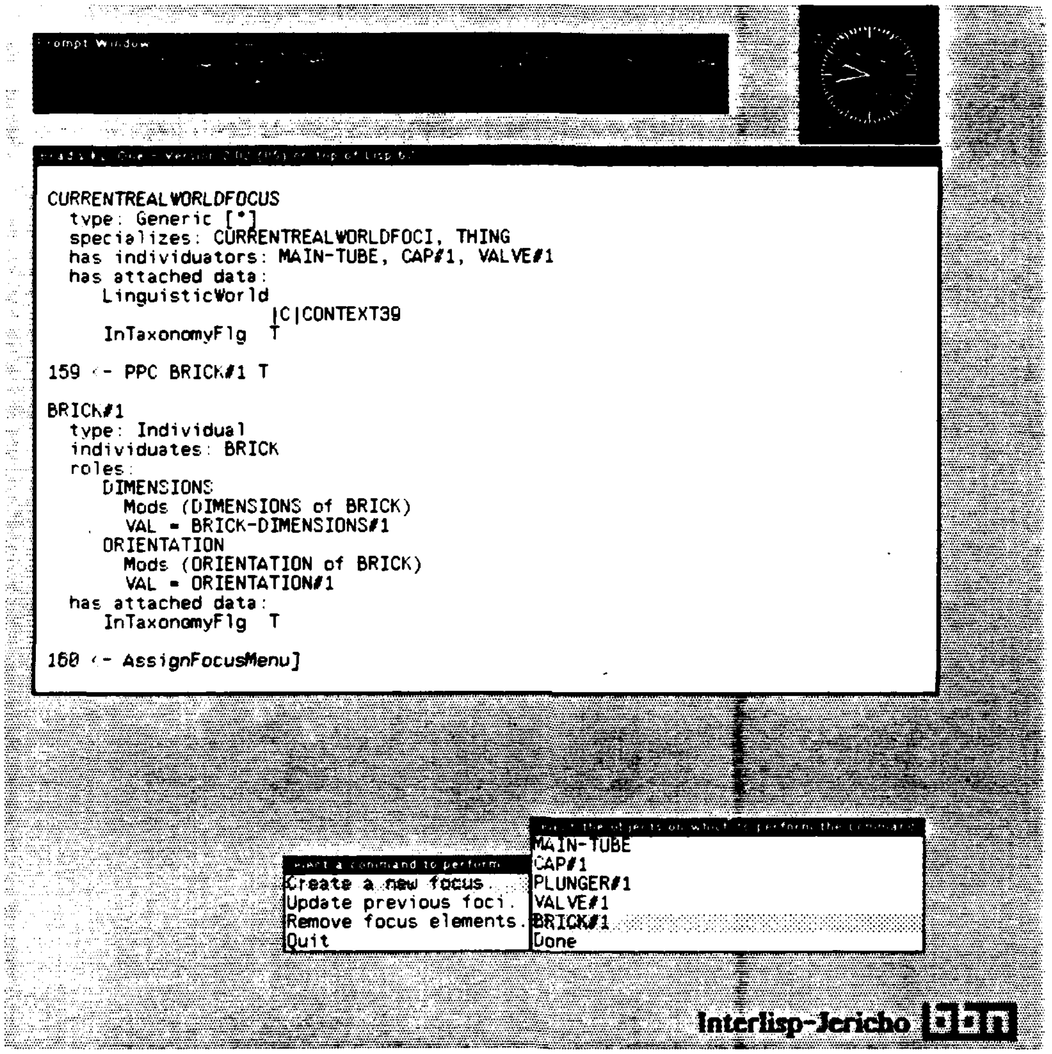
Throughout the above discussion, the shifting of focus was described from the point of view of the real world, i.e., by describing which objects in the world are partitioned into a particular focus space. I never mentioned, however, that a corresponding shift was also occurring to partitions of the linguistic world. The linguistic world is partitioned in accordance with the real world partitions to make reference resolution more efficient and so that anaphoric references can be resolved. It makes referent identification simpler because the most relevant objects can be checked first. It allows for the resolution of anaphoric definite noun phrases because the linguistic world contains a conglomeration of previous references to an object. If the current input fits in line with the previous ones (i.e., there are no discrepancies), then the anaphoric expression is assumed to refer to the same real world object as the previous ones.

Figure C-3 shows my actual focus mechanism put through its paces.



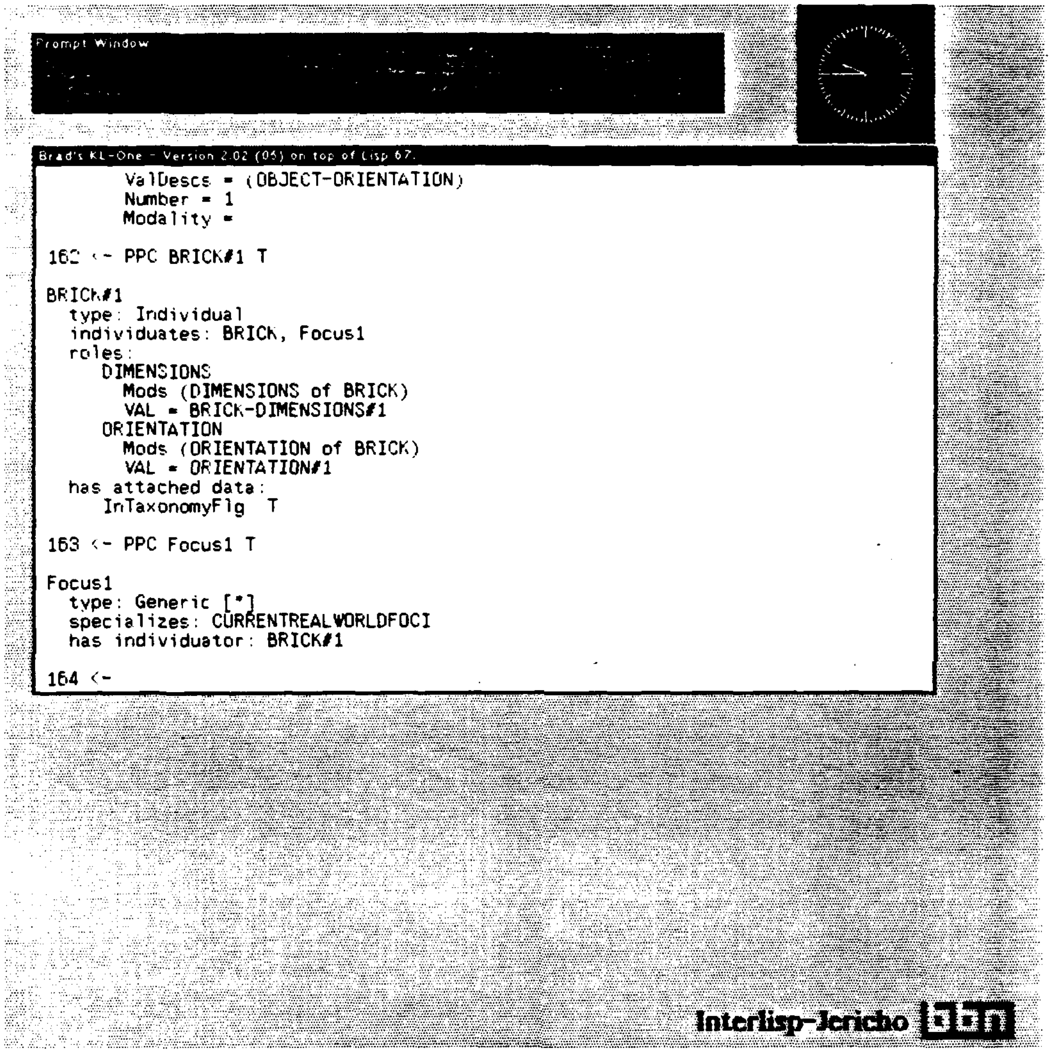
```
  has individuators: PLUNGER#1, MAIN-TUBE, CAP#1, VALVE#1
  has attached data:
     InTaxonomyFlg  T

157 <- PPC REALWORLDCATCHALLFOCUS

REALWORLDCATCHALLFOCUS
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI, THING
  has individuator: PLUNGER#1
  has attached data:
     LinguisticWorld
                   |C|CATCHALLCONTEXT
     InTaxonomyFlg  T

158 <- PPC CURRENTREALWORLDFOCUS

CURRENTREALWORLDFOCUS
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI, THING
  has individuators: MAIN-TUBE, CAP#1, VALVE#1
  has attached data:
     LinguisticWorld
                   |C|CONTEXT39
     InTaxonomyFlg  T

159 <-
```

Figure C-3: Using the focus mechanism



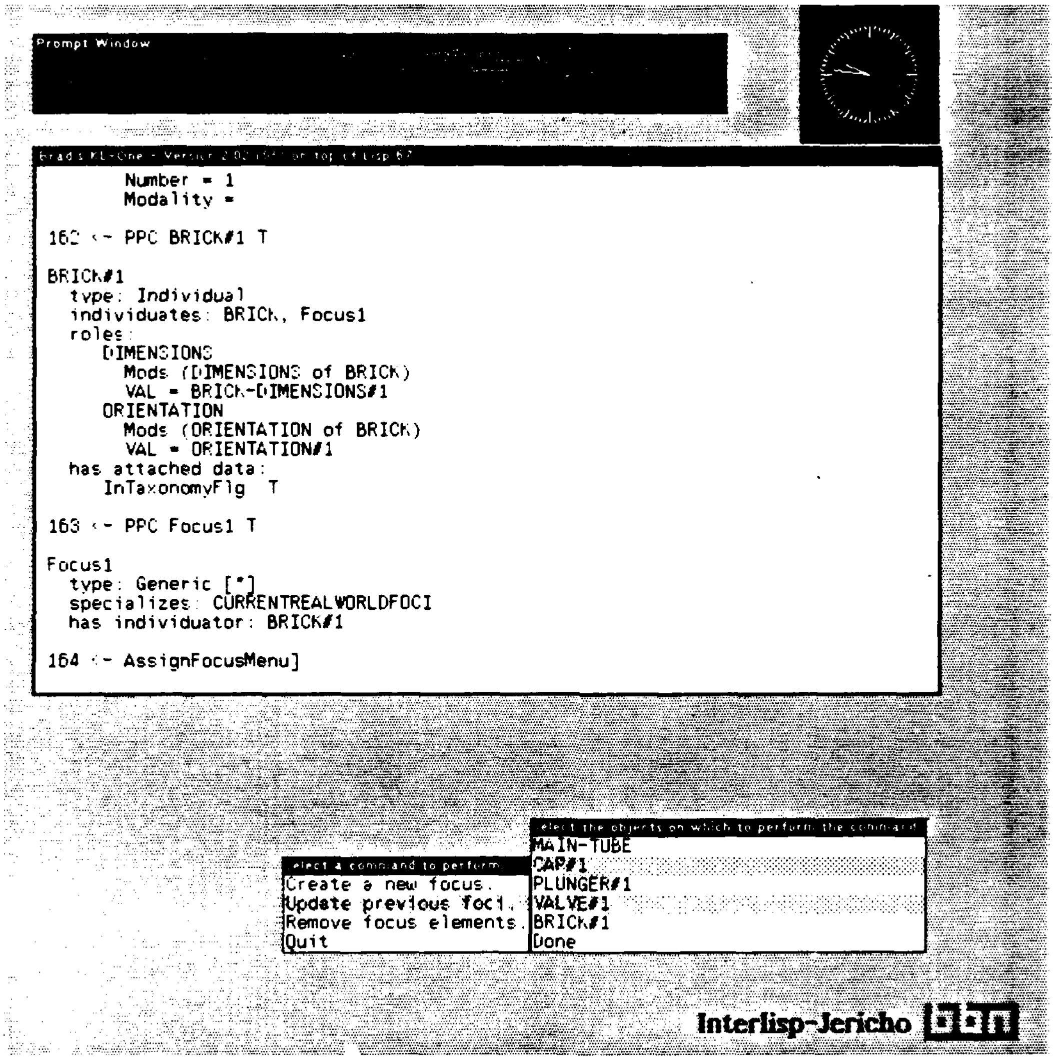
```
CURRENTREALWORLDFOCUS
  tvpe: Generic [*]
  specializes: CURRENTREALWORLDFOCI, THING
  has individuators: MAIN-TUBE, CAP#1, VALVE#1
  has attached data:
     LinguisticWorld
                   |C|CONTEXT39
     InTaxonomyFlg  T

159 <- PPC BRICK#1 T

BRICK#1
  type: Individual
  individuates: BRICK
  roles:
     DIMENSIONS
       Mods (DIMENSIONS of BRICK)
       VAL = BRICK-DIMENSIONS#1
     ORIENTATION
       Mods (ORIENTATION of BRICK)
       VAL = ORIENTATION#1
  has attached data:
     InTaxonomyFlg  T

160 <- AssignFocusMenu]
```

FIGURE C-3, CONTINUED



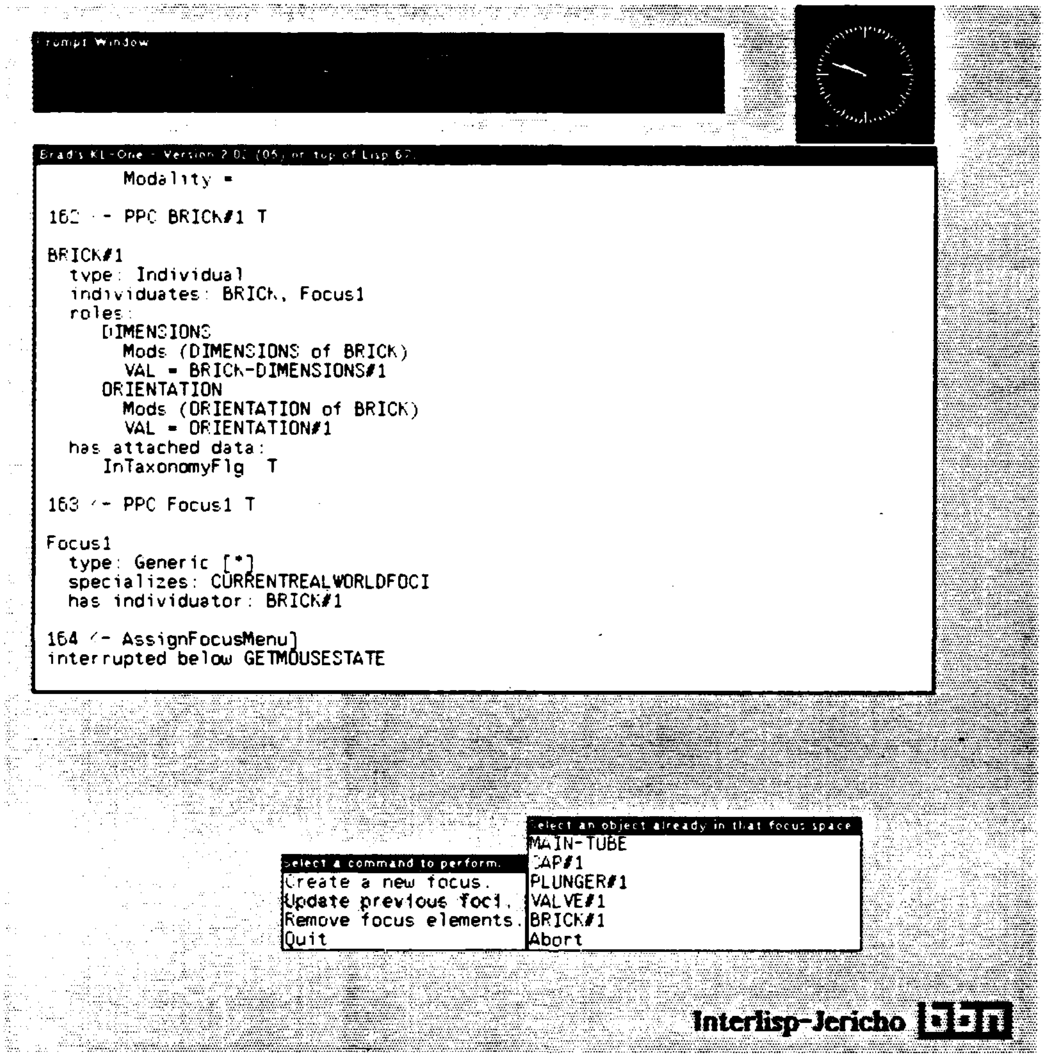
```
       ValDescs = (OBJECT-ORIENTATION)
       Number = 1
       Modality =

162 <- PPC BRICK#1 T

BRICK#1
  type: Individual
  individuates: BRICK, Focus1
  roles:
     DIMENSIONS
       Mods (DIMENSIONS of BRICK)
       VAL = BRICK-DIMENSIONS#1
     ORIENTATION
       Mods (ORIENTATION of BRICK)
       VAL = ORIENTATION#1
  has attached data:
     InTaxonomyFlg  T

163 <- PPC Focus1 T

Focus1
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuator: BRICK#1

164 <-
```

FIGURE C-3, CONTINUED



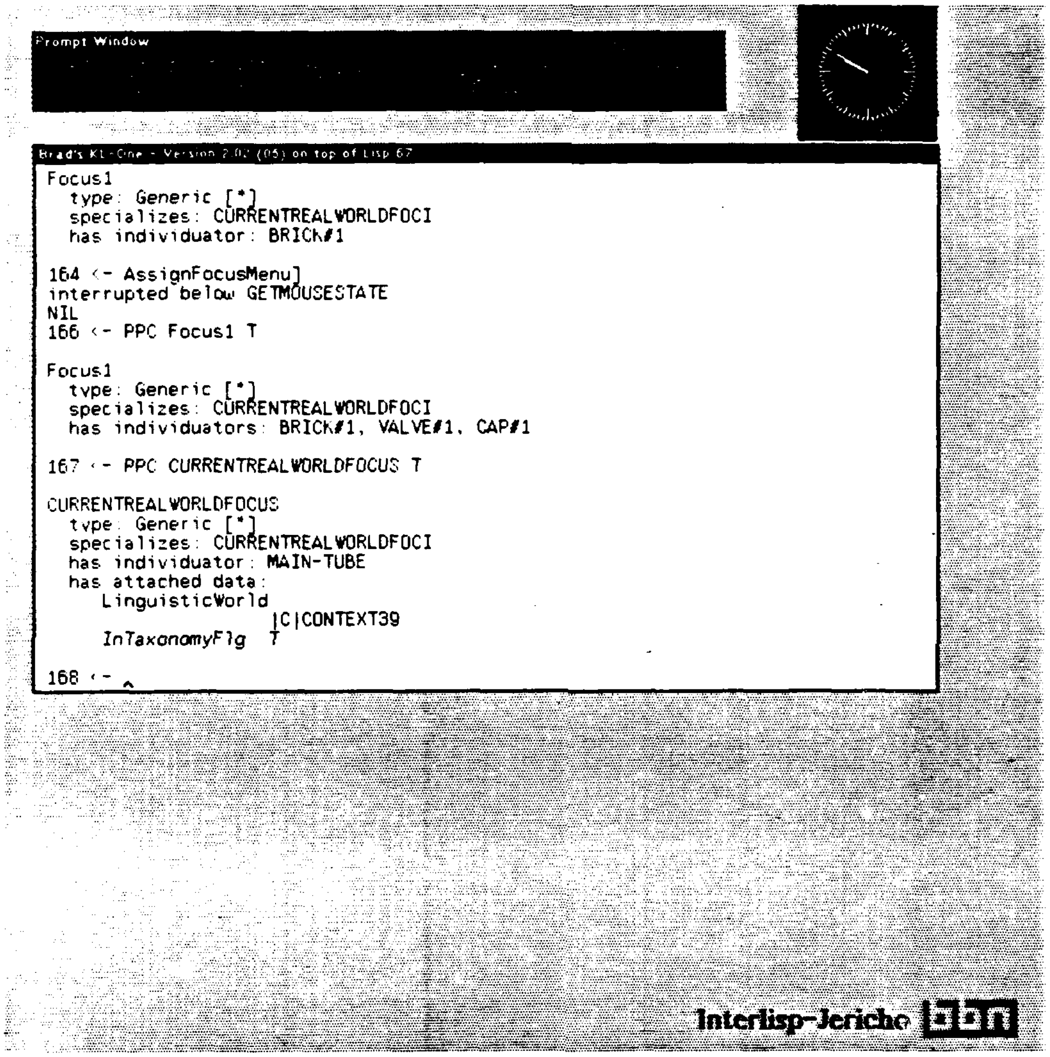
```
        Number = 1
        Modality =

 162 <- PPC BRICK#1 T

 BRICK#1
   type: Individual
   individuates: BRICK, Focus1
   roles:
      DIMENSIONS
        Mods (DIMENSIONS of BRICK)
        VAL = BRICK-DIMENSIONS#1
      ORIENTATION
        Mods (ORIENTATION of BRICK)
        VAL = ORIENTATION#1
   has attached data:
      InTaxonomyFlg  T

 163 <- PPC Focus1 T

 Focus1
   type: Generic [*]
   specializes: CURRENTREALWORLDFOCI
   has individuator: BRICK#1

 164 <- AssignFocusMenu]
```

FIGURE C-3, CONTINUED



```
       Modality =

162 - PPC BRICK#1 T

BRICK#1
  type: Individual
  individuates: BRICK, Focus1
  roles:
     DIMENSIONS
       Mods (DIMENSIONS of BRICK)
       VAL = BRICK-DIMENSIONS#1
     ORIENTATION
       Mods (ORIENTATION of BRICK)
       VAL = ORIENTATION#1
  has attached data:
     InTaxonomyFlg  T

163 ← PPC Focus1 T

Focus1
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuator: BRICK#1

164 ← AssignFocusMenu]
interrupted below GETMOUSESTATE
```

FIGURE C-3, CONTINUED

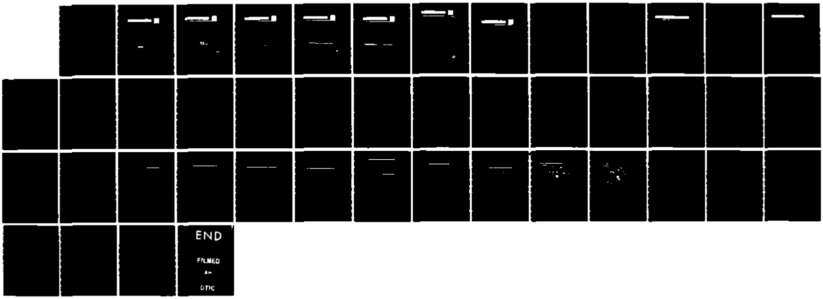


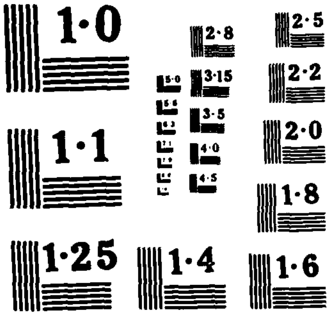
```
Focus1
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuator: BRICK#1

164 <- AssignFocusMenu]
interrupted below GETMOUSESTATE
NIL
166 <- PPC Focus1 T

Focus1
  tvpe: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuators: BRICK#1, VALVE#1, CAP#1

167 <- PPC CURRENTREALWORLDFOCUS T

CURRENTREALWORLDFOCUS
  tvpe: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuator: MAIN-TUBE
  has attached data:
     LinguisticWorld
                    |C|CONTEXT39
     InTaxonomyFlg  T

168 <- ^
```

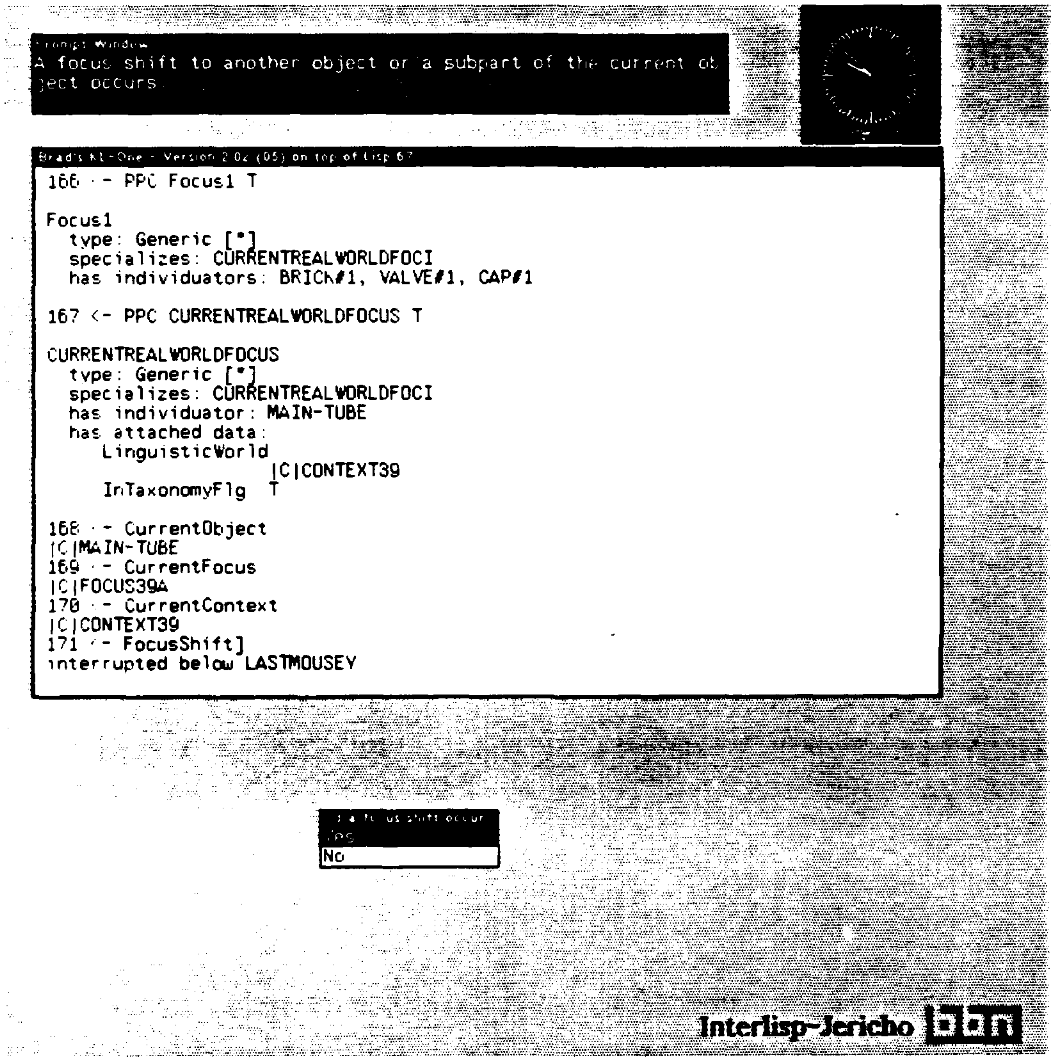
FIGURE C-3, CONTINUED

AD-A162 843 COMMUNICATION AND MISCOMMUNICATION(U) BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA B A GOODMAN OCT 85 BBN-5681 N00014-85-C-0079 3/3

UNCLASSIFIED F/G 9/2 NL



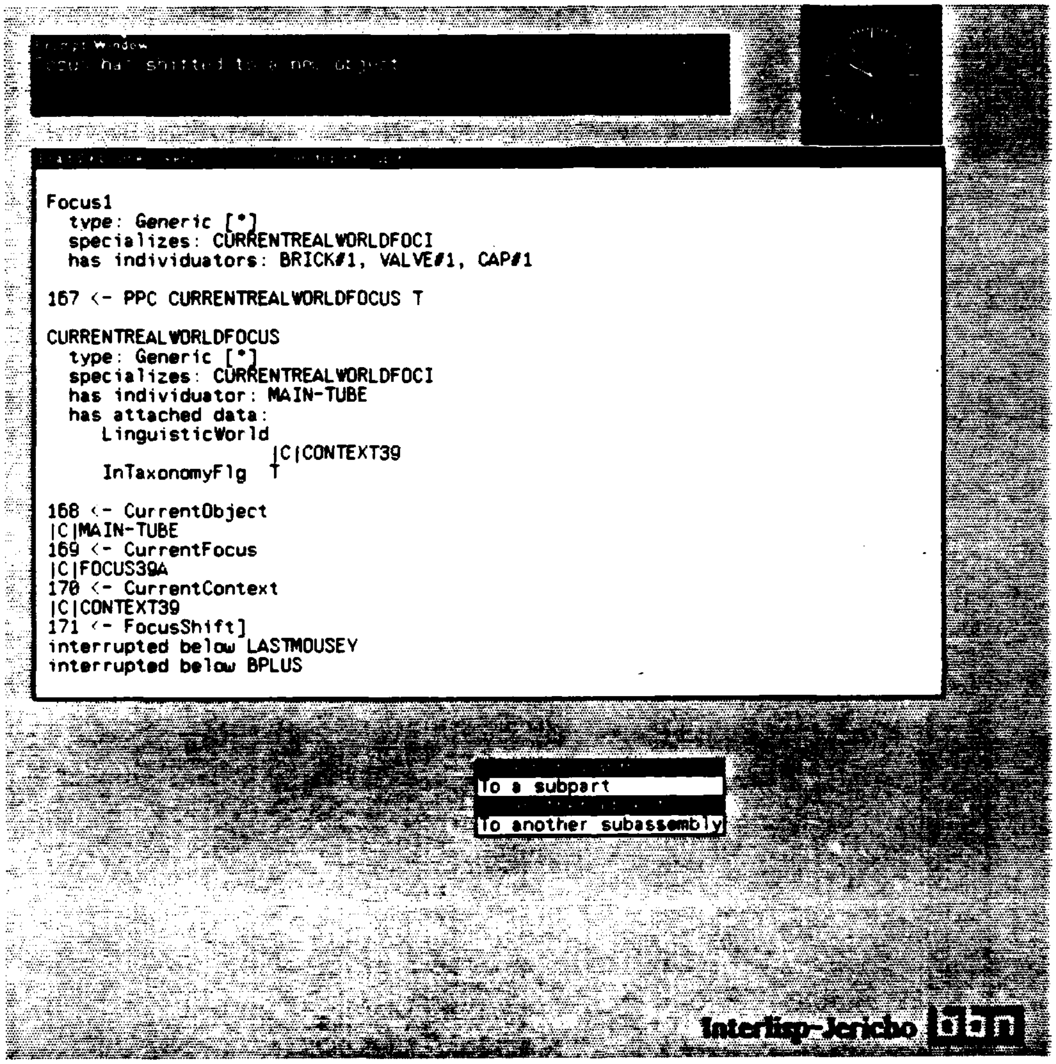

1·0
2·8
2·5
5·0
3·15
2·2
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6



A focus shift to another object or a subpart of the current object occurs.

```
166 <- PPC Focus1 T

Focus1
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuators: BRICK#1, VALVE#1, CAP#1

167 <- PPC CURRENTREALWORLDFOCUS T

CURRENTREALWORLDFOCUS
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuator: MAIN-TUBE
  has attached data:
     LinguisticWorld
                    |C|CONTEXT39
     InTaxonomyFlg  T

168 <- CurrentObject
|C|MAIN-TUBE
169 <- CurrentFocus
|C|FOCUS39A
170 <- CurrentContext
|C|CONTEXT39
171 <- FocusShift]
interrupted below LASTMOUSEY
```

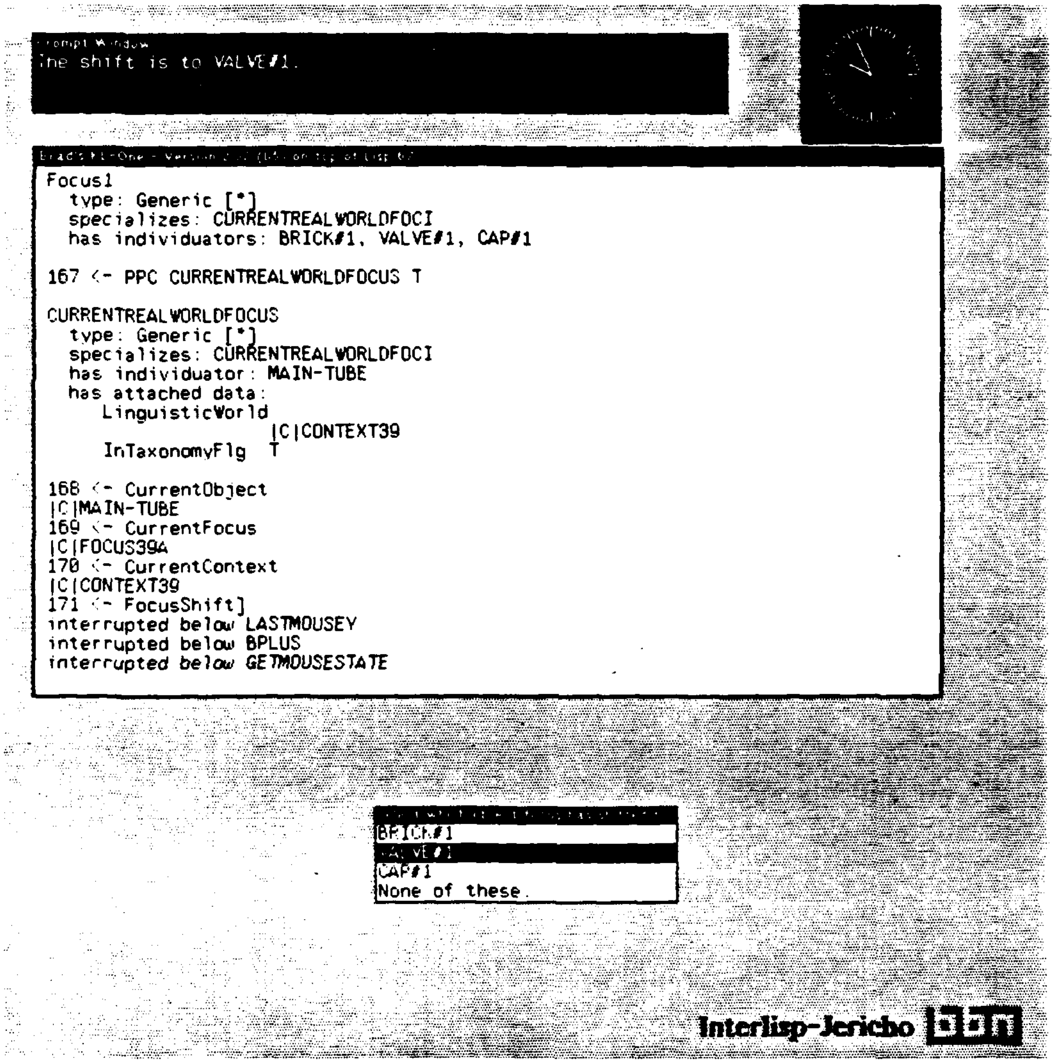
FIGURE C-3, CONTINUED



Focus has shifted to a new object

```
Focus1
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuators: BRICK#1, VALVE#1, CAP#1

167 <- PPC CURRENTREALWORLDFOCUS T

CURRENTREALWORLDFOCUS
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuator: MAIN-TUBE
  has attached data:
     LinguisticWorld
                    |C|CONTEXT39
     InTaxonomyFlg  T

168 <- CurrentObject
|C|MAIN-TUBE
169 <- CurrentFocus
|C|FOCUS39A
170 <- CurrentContext
|C|CONTEXT39
171 <- FocusShift]
interrupted below LASTMOUSEY
interrupted below BPLUS
```

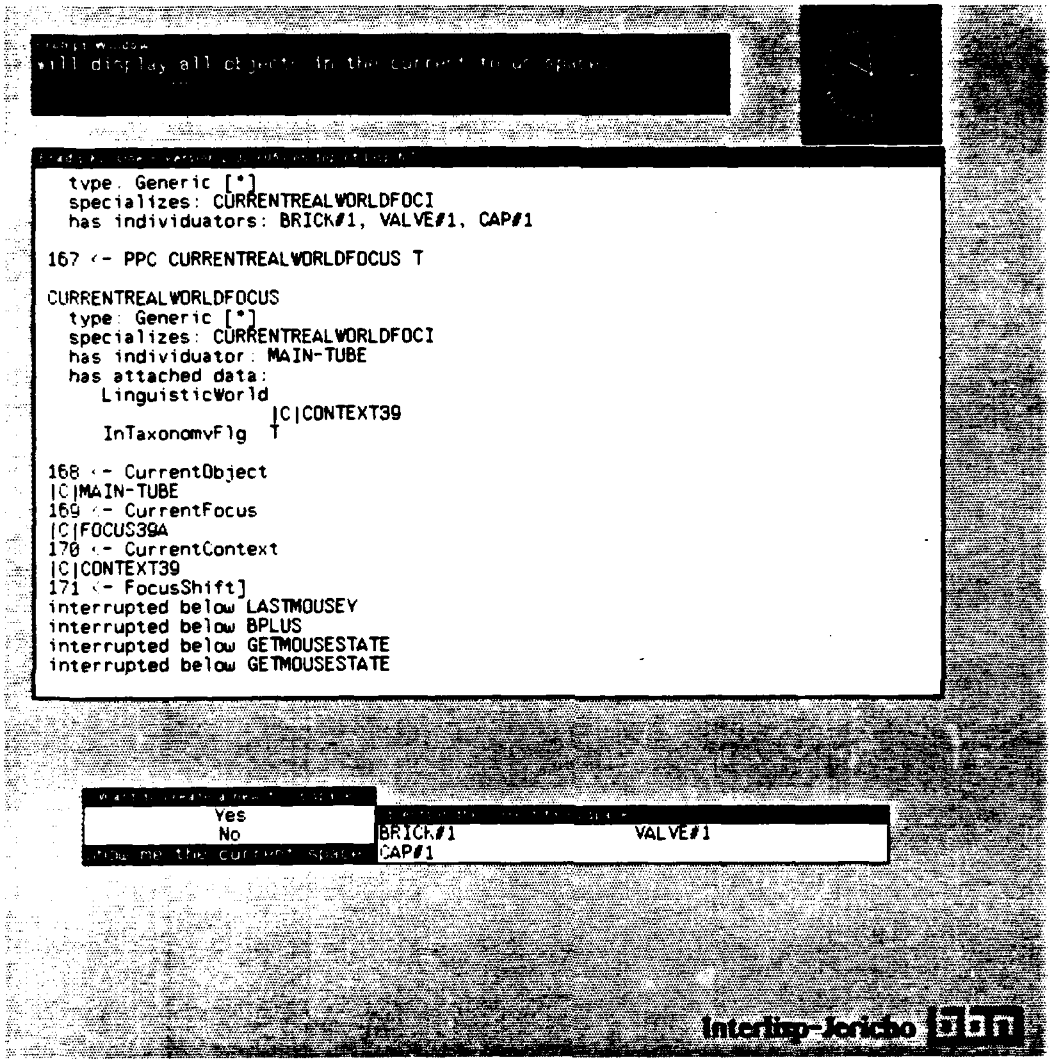
FIGURE C-3. CONTINUED



```
The shift is to VALVE#1.
```

```
Focus1
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuators: BRICK#1, VALVE#1, CAP#1

167 <- PPC CURRENTREALWORLDFOCUS T

CURRENTREALWORLDFOCUS
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuator: MAIN-TUBE
  has attached data:
     LinguisticWorld
                    |C|CONTEXT39
     InTaxonomyFlg  T

168 <- CurrentObject
|C|MAIN-TUBE
169 <- CurrentFocus
|C|FOCUS39A
170 <- CurrentContext
|C|CONTEXT39
171 <- FocusShift]
interrupted below LASTMOUSEY
interrupted below BPLUS
interrupted below GETMOUSESTATE
```

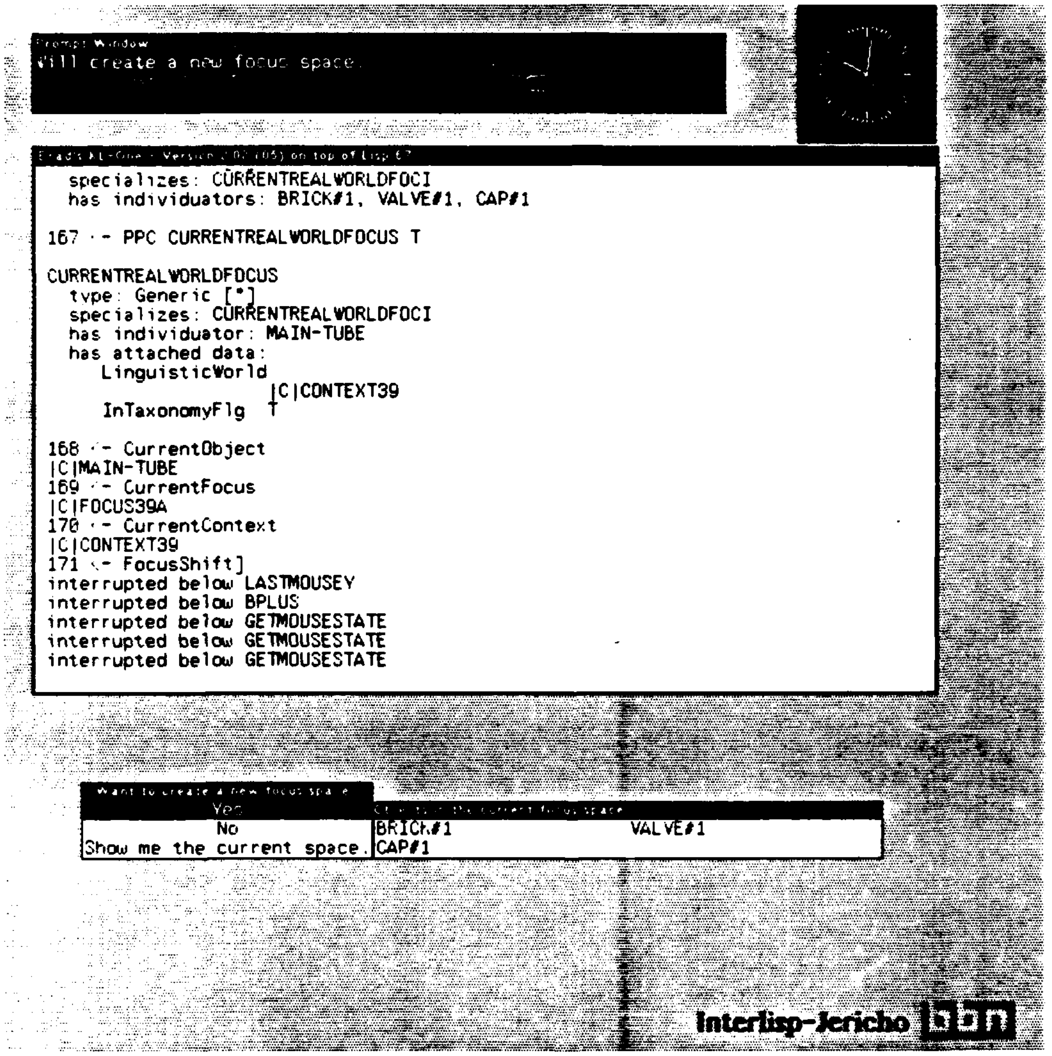
FIGURE C-3, CONTINUED



```
will display all objects in the current focus space.
```

```
  tvpe. Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuators: BRICK#1, VALVE#1, CAP#1

167 <- PPC CURRENTREALWORLDFOCUS T

CURRENTREALWORLDFOCUS
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuator: MAIN-TUBE
  has attached data:
     LinguisticWorld
                    |C|CONTEXT39
     InTaxonomvFlg  T

168 <- CurrentObject
|C|MAIN-TUBE
169 <- CurrentFocus
|C|FOCUS39A
170 <- CurrentContext
|C|CONTEXT39
171 <- FocusShift]
interrupted below LASTMOUSEY
interrupted below BPLUS
interrupted below GETMOUSESTATE
interrupted below GETMOUSESTATE
```

FIGURE C-3, CONTINUED

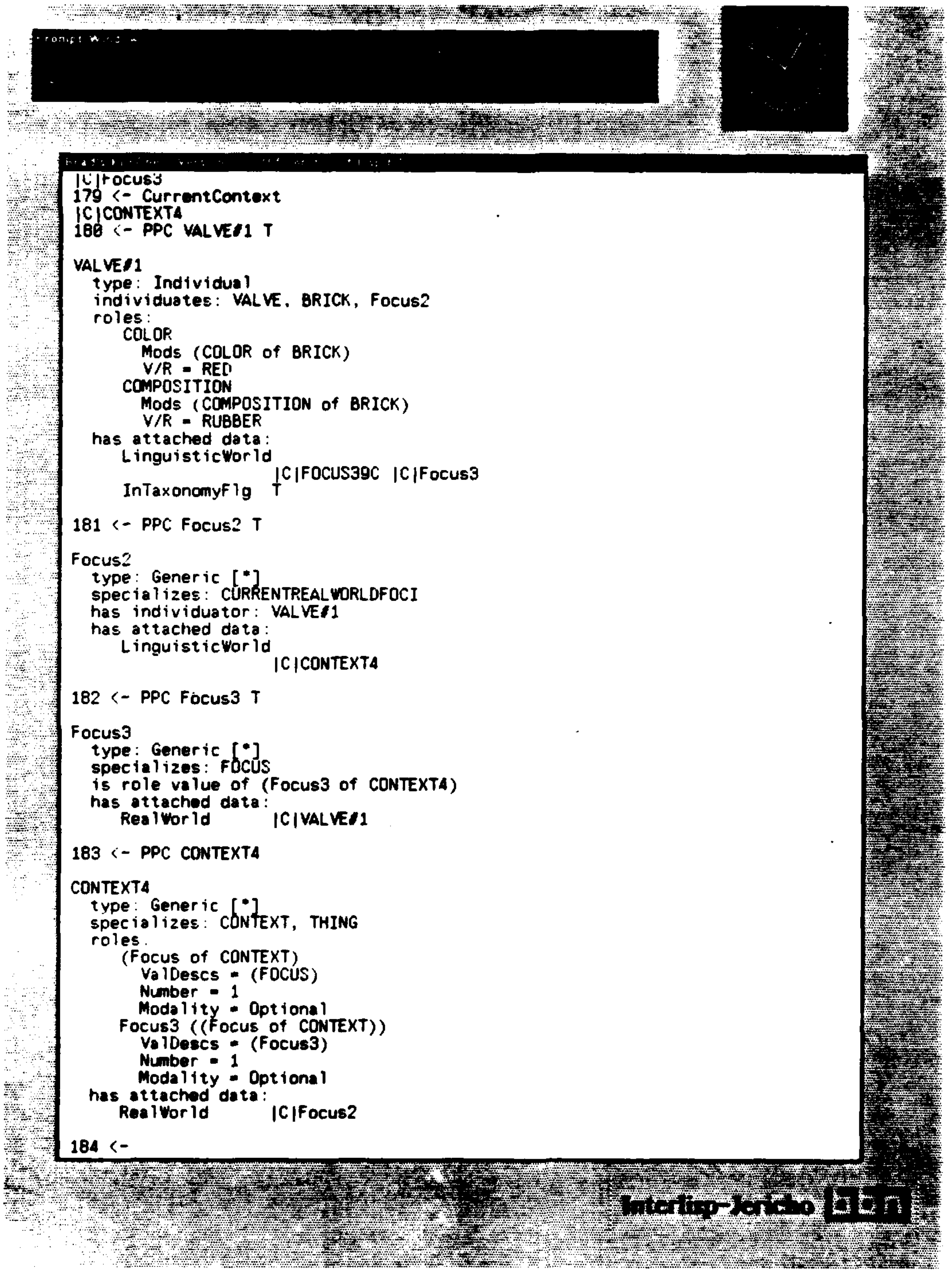


```
Will create a new focus space.
```

```
  specializes: CURRENTREALWORLDFOCI
  has individuators: BRICK#1, VALVE#1, CAP#1

167 ·- PPC CURRENTREALWORLDFOCUS T

CURRENTREALWORLDFOCUS
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuator: MAIN-TUBE
  has attached data:
     LinguisticWorld
                    |C|CONTEXT39
     InTaxonomyFlg  T
168 ·- CurrentObject
|C|MAIN-TUBE
169 ·- CurrentFocus
|C|FOCUS39A
170 ·- CurrentContext
|C|CONTEXT39
171 ·- FocusShift]
interrupted below LASTMOUSEY
interrupted below BPLUS
interrupted below GETMOUSESTATE
interrupted below GETMOUSESTATE
interrupted below GETMOUSESTATE
```

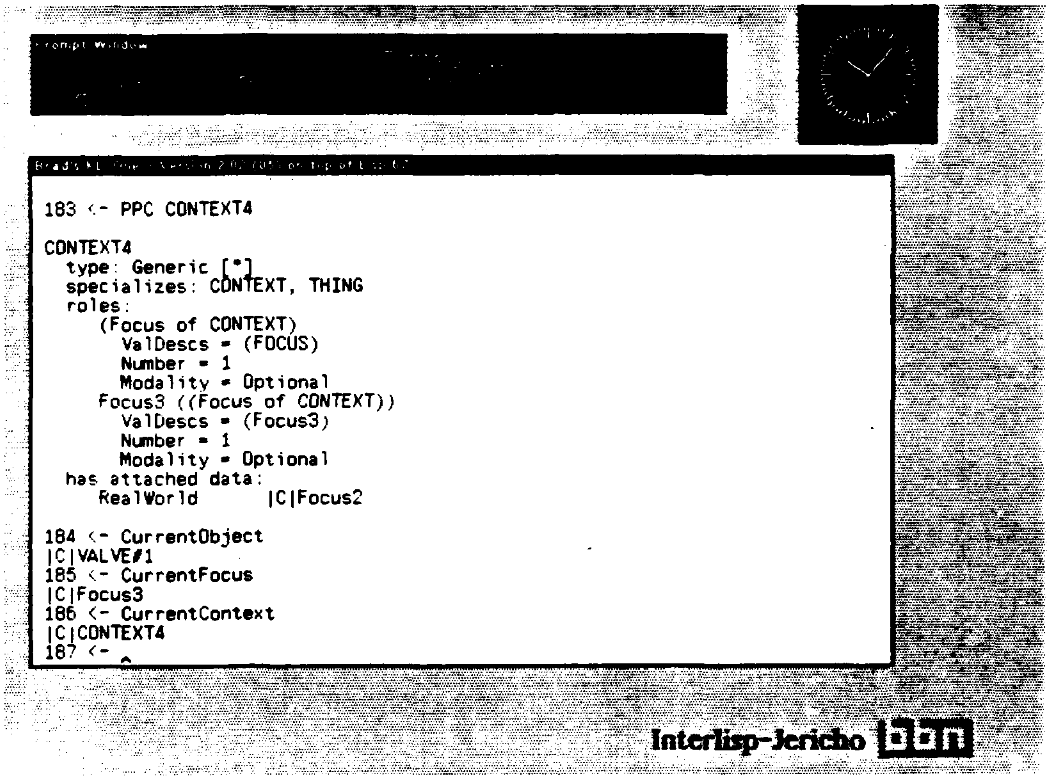
BRICK#1	VALVE#1
CAP#1	

FIGURE C-3, CONTINUED



```
|C|Focus3
179 <- CurrentContext
|C|CONTEXT4
180 <- PPC VALVE#1 T

VALVE#1
  type: Individual
  individuates: VALVE, BRICK, Focus2
  roles:
     COLOR
       Mods (COLOR of BRICK)
       V/R = RED
     COMPOSITION
       Mods (COMPOSITION of BRICK)
       V/R = RUBBER
  has attached data:
     LinguisticWorld
                    |C|FOCUS39C |C|Focus3
     InTaxonomyFlg  T

181 <- PPC Focus2 T

Focus2
  type: Generic [*]
  specializes: CURRENTREALWORLDFOCI
  has individuator: VALVE#1
  has attached data:
     LinguisticWorld
                    |C|CONTEXT4

182 <- PPC Focus3 T

Focus3
  type: Generic [*]
  specializes: FOCUS
  is role value of (Focus3 of CONTEXT4)
  has attached data:
     RealWorld      |C|VALVE#1

183 <- PPC CONTEXT4

CONTEXT4
  type: Generic [*]
  specializes: CONTEXT, THING
  roles.
     (Focus of CONTEXT)
       ValDescs = (FOCUS)
       Number = 1
       Modality = Optional
     Focus3 ((Focus of CONTEXT))
       ValDescs = (Focus3)
       Number = 1
       Modality = Optional
  has attached data:
     RealWorld      |C|Focus2

184 <-
```

FIGURE C-3, CONTINUED

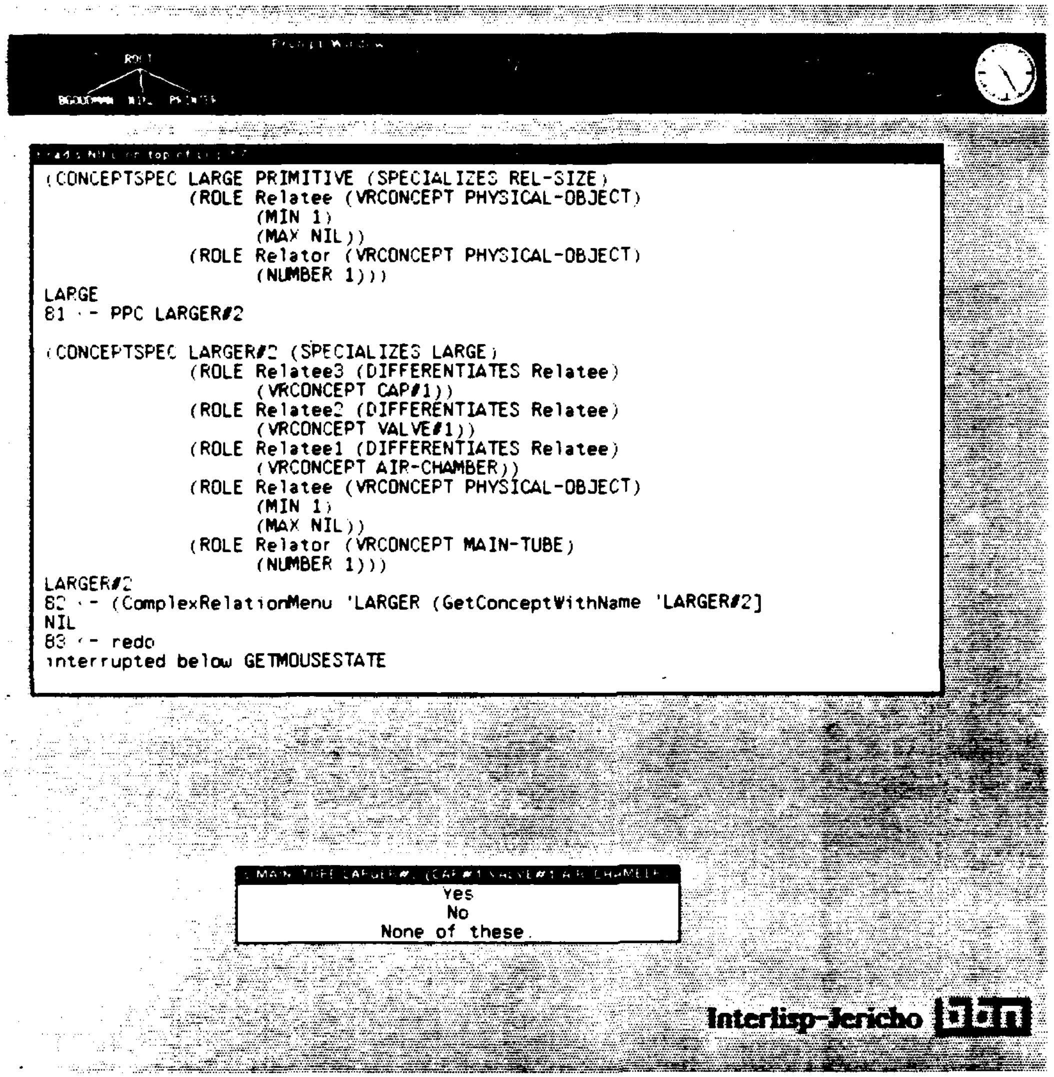


```
183 <- PPC CONTEXT4

CONTEXT4
  type: Generic [*]
  specializes: CONTEXT, THING
  roles:
     (Focus of CONTEXT)
       ValDescs = (FOCUS)
       Number = 1
       Modality = Optional
     Focus3 ((Focus of CONTEXT))
       ValDescs = (Focus3)
       Number = 1
       Modality = Optional
  has attached data:
     RealWorld      |C|Focus2

184 <- CurrentObject
|C|VALVE#1
185 <- CurrentFocus
|C|Focus3
186 <- CurrentContext
|C|CONTEXT4
187 <-
```

FIGURE C-3. CONCLUDED

APPENDIX D
HANDLING COMPARATIVES, SUPERLATIVES, AND COMPLEX RELATIONS

This appendix shows the menu-driven mechanism used to simulate the use of comparatives, superlatives, and complex relations in a speaker's description. Figure D-1 shows the knowledge base representation of such relations and the use of menus by the system to interact with the user to determine if the relation is satisfied.



```
(CONCEPTSPEC LARGE PRIMITIVE (SPECIALIZES REL-SIZE)
              (ROLE Relatee (VRCONCEPT PHYSICAL-OBJECT)
                    (MIN 1)
                    (MAX NIL))
              (ROLE Relator (VRCONCEPT PHYSICAL-OBJECT)
                    (NUMBER 1)))
LARGE
81 ←- PPC LARGER#2

(CONCEPTSPEC LARGER#2 (SPECIALIZES LARGE)
              (ROLE Relatee3 (DIFFERENTIATES Relatee)
                    (VRCONCEPT CAP#1))
              (ROLE Relatee2 (DIFFERENTIATES Relatee)
                    (VRCONCEPT VALVE#1))
              (ROLE Relatee1 (DIFFERENTIATES Relatee)
                    (VRCONCEPT AIR-CHAMBER))
              (ROLE Relatee (VRCONCEPT PHYSICAL-OBJECT)
                    (MIN 1)
                    (MAX NIL))
              (ROLE Relator (VRCONCEPT MAIN-TUBE)
                    (NUMBER 1)))
LARGER#2
82 ←- (ComplexRelationMenu 'LARGER (GetConceptWithName 'LARGER#2]
NIL
83 ←- redo
interrupted below GETMOUSESTATE
```

Figure D-1: Handling comparatives

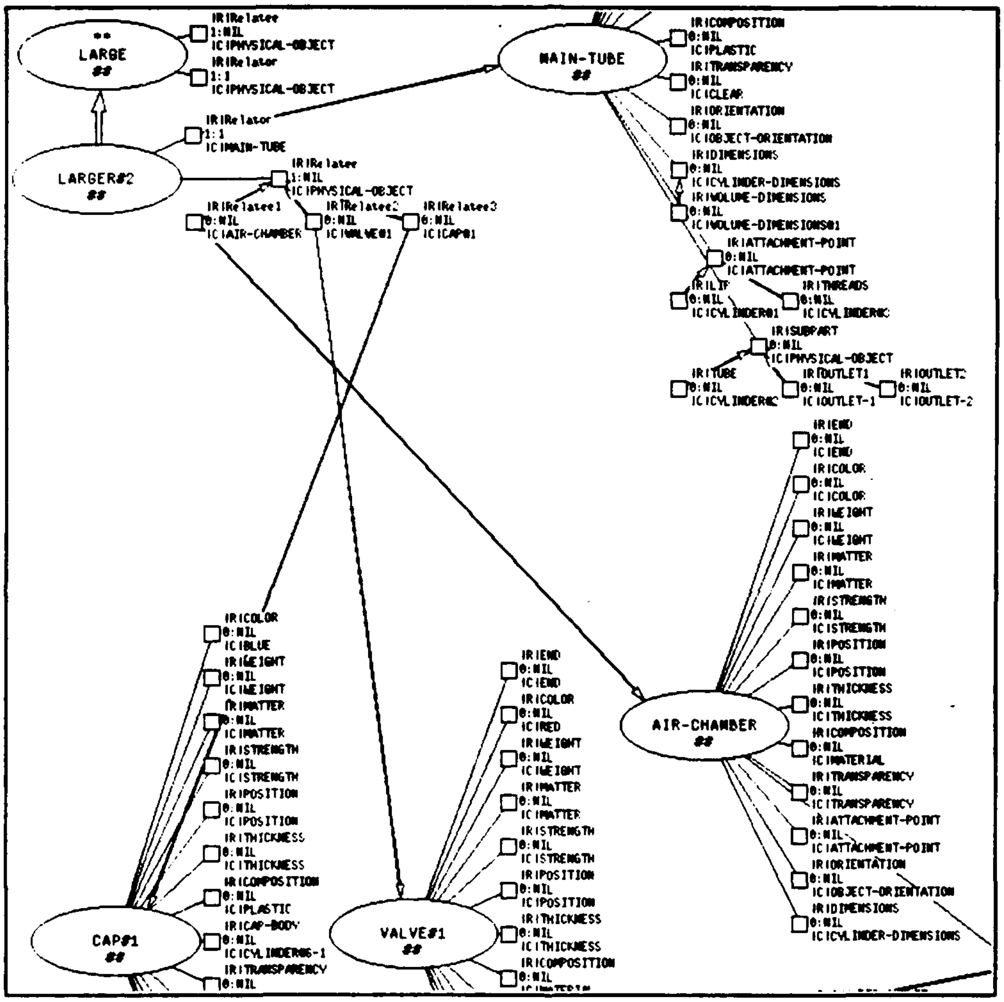
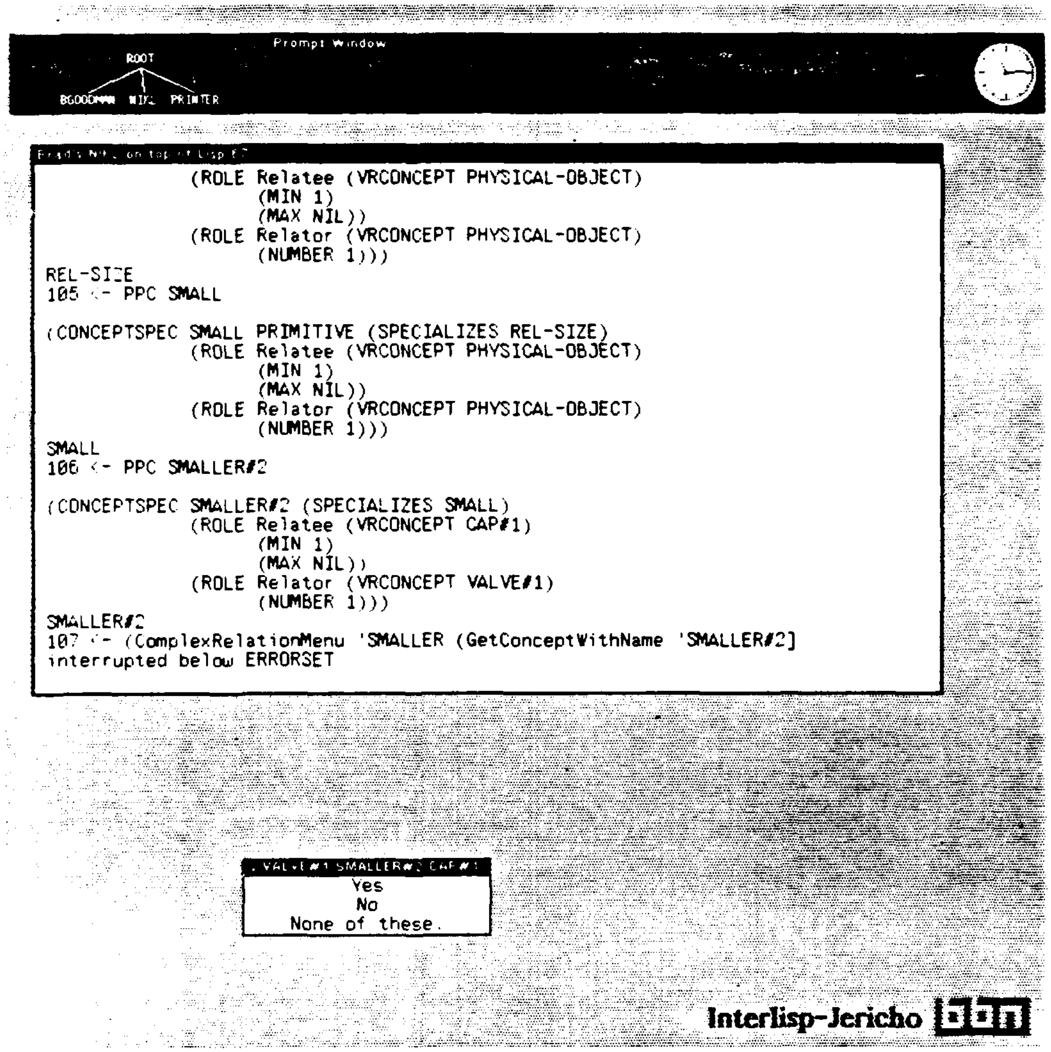


FIGURE D-1, CONTINUED



```
               (ROLE Relatee (VRCONCEPT PHYSICAL-OBJECT)
                      (MIN 1)
                      (MAX NIL))
               (ROLE Relator (VRCONCEPT PHYSICAL-OBJECT)
                      (NUMBER 1)))
REL-SIZE
105 <- PPC SMALL

(CONCEPTSPEC SMALL PRIMITIVE (SPECIALIZES REL-SIZE)
               (ROLE Relatee (VRCONCEPT PHYSICAL-OBJECT)
                      (MIN 1)
                      (MAX NIL))
               (ROLE Relator (VRCONCEPT PHYSICAL-OBJECT)
                      (NUMBER 1)))
SMALL
106 <- PPC SMALLER#2

(CONCEPTSPEC SMALLER#2 (SPECIALIZES SMALL)
               (ROLE Relatee (VRCONCEPT CAP#1)
                      (MIN 1)
                      (MAX NIL))
               (ROLE Relator (VRCONCEPT VALVE#1)
                      (NUMBER 1)))
SMALLER#2
107 <- (ComplexRelationMenu 'SMALLER (GetConceptWithName 'SMALLER#2]
interrupted below ERRORSET
```

FIGURE D-1, CONTINUED

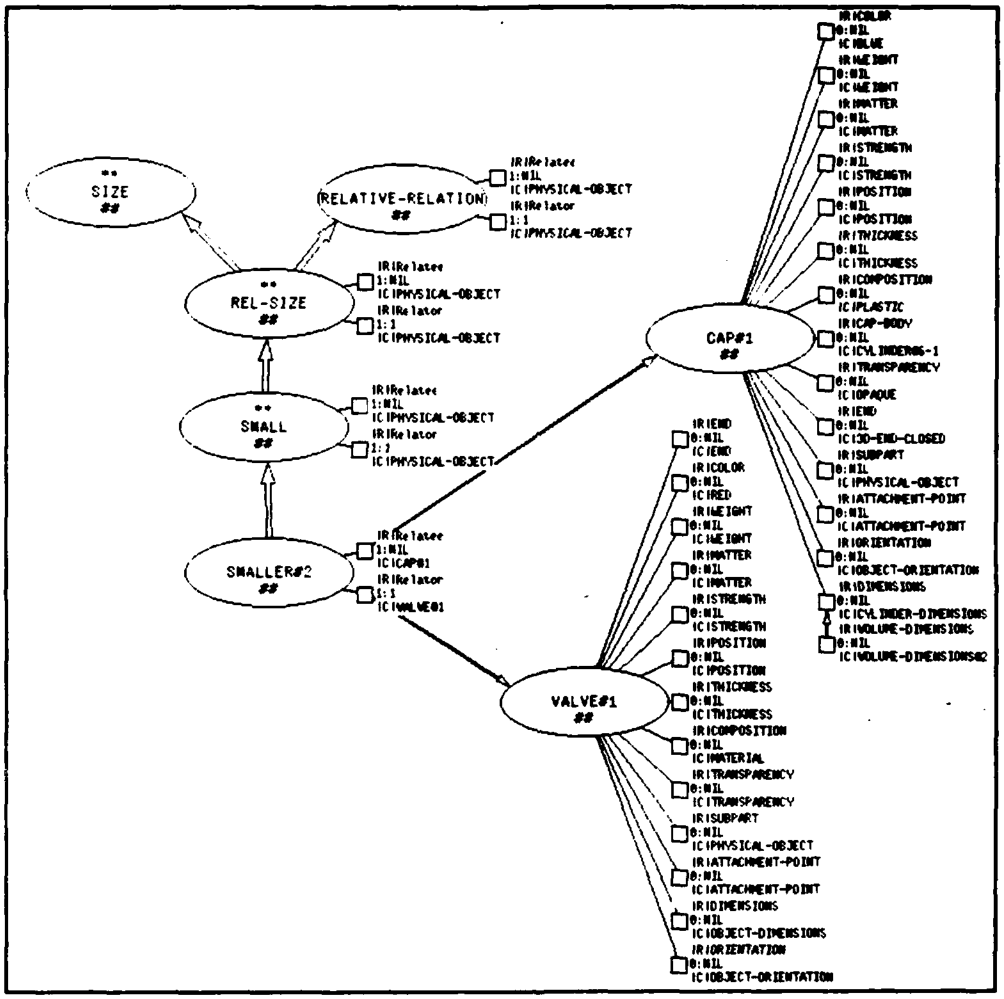


FIGURE D-1, CONCLUDED

APPENDIX E
THE BASIC REFERENCE MECHANISM IN ACTION

This appendix provides a commented trace of the reference mechanism in action. It shows in Figure E-1 what happens when no referent is found initially. A walk in the taxonomy and the use of the partial matcher help select referent candidates.

```
55 <- (* Start of demonstration of reference identification
         mechanism with focus mechanism.  Will manually run
         it through, highlighting important aspects.

         Will start off by showing relevant parts of the
         taxonomy at the beginning.)

56 <- PPC TUBE1 T

TUBE1
  type: Generic
  specializes: TUBE, FOCUS39A
  roles:
     COLOR
       Mods (COLOR of TUBE)
       V/R = VIOLET

57 <- PPC CONTEXT39 T

CONTEXT39
  type: Generic [*]
  specializes: CONTEXT
  roles:
     Focus1
       Diffs (Focus of CONTEXT)
       V/R = FOCUS39A
     Focus2
       Diffs (Focus of CONTEXT)
       V/R = FOCUS39B
     Focus3
       Diffs (Focus of CONTEXT)
       V/R = FOCUS39C
  has attached data:
     RealWorld        |C|CURRENTREALWORLDFOCUS

58 <- PPC FOCUS39A

FOCUS39A
  type: Generic [*]
  specializes: FOCUS, THING
  has specializer: TUBE1
  roles:
     (SubFocus of FOCUS)
       ValDescs = (FOCUS)
       Number = 1
       Modality = Optional
     SubFocus39A ((SubFocus of FOCUS))
       ValDescs = (SUBFOCUS39A)
       Number = 1
       Modality = Optional
  is role value of (Focus1 of CONTEXT39)
  has attached data:
     RealWorld        |C|MAIN-TUBE
```

Figure E-1: Sample run of the basic reference mechanism

```
60 <- (* Now will mark all current concepts as being in the taxonomy.
        This is done so that the Classifier knows the concepts are
        present.)

61 <- AddAllConceptsToTaxonomy]
|C|DEVICE |C|Action |C|Agent |C|PUT |C|DEFAULT |C|SHAPE |C|ITEMIZE-ACT
|C|CONNECT-ACT |C|FIT |C|FIT-ONTO |C|MOVEMENT-ACT |C|FIT-INTO
|C|PUT-ONTO |C|PUT-INTO |C|PLACE-ONTO |C|PLACE-INTO |C|END
|C|INSERT-ONTO |C|INSERT-INTO |C|PUSH-INTO |C|PUSH-ONTO |C|TWIST
|C|TWIST-ONTO |C|TWIST-INTO |C|SCREW |C|SCREW-ONTO |C|SCREW-INTO
|C|ABSTRACTION |C|CAP#1 |C|VALVE#1 |C|PLUNGER |C|CYLINDER#1
|C|CYLINDER#2 |C|POSITION |C|CYLINDER#4 |C|CYLINDER#5 |C|COLOR
|C|CYLINDER#6 |C|PLUNGER#1 |C|CYLINDER#3 |C|DIMENSIONS#2 |C|FOCUS
|C|ORIENTATION#2 |C|DIMENSIONS#3 |C|ORIENTATION#3 |C|INSERT
|C|DIMENSIONS#4 |C|FLOATP |C|ORIENTATION#4 |C|DIMENSIONS#5
|C|ORIENTATION#5 |C|CURRENTREALWORLDFOCI |C|DIMENSIONS#6
|C|REALWORLDCATCHALLFOCUS |C|ORIENTATION#6 |C|CURRENTREALWORLDFOCUS
|C|STRINGP |C|OPAQUE |C|DIMENSIONS#7 |C|SMALLP |C|ORIENTATION#7
|C|FOCUS39A |C|FIXP |C|ROTATION#2 |C|ATOM |C|NUMBERP |C|TRANSLATION#2
|C|SUBFOCUS39A |C|LITATOM  |C|ROTATION#3 |C|FOCUS39B |C|LISTP
|C|TRANSLATION#3 |C|FOCUS39C |C|ROTATION#4 |C|TUBE1 |C|CLEAR |C|TUBE4
|C|ATTACH |C|ROTATION#5 |C|TUBE5 |C|TRANSLATION#5 |C|CAP1 |C|WEIGHT
|C|ROTATION#6 |C|TRANSLATION#6 |C|CONTEXT |C|ROTATION |C|ROTATION#7
|C|TRANSLATION#4 |C|TRANSLATION#7 |C|CATCHALLCONTEXT |C|FOCUS1
|C|FOCUS2 |C|BRICK#1 |C|PLUNGER66 |C|BRICK-DIMENSIONS#1 |C|TUBE77
|C|ORIENTATION#1 |C|TUBE88 |C|ROTATION#1 |C|CONTEXT39 |C|TRANSLATION#1
|C|MAIN-TUBE |C|TRANSLATION |C|CAP |C|PLACE |C|PHYSICAL-OBJECT
|C|MEANING-UNIT |C|MATERIAL |C|PHYSICAL-PROPERTY |C|TRANSPARENCY
|C|STRENGTH |C|MATTER |C|OBJECT-DIMENSIONS |C|OBJECT-ORIENTATION
|C|THICKNESS |C|PUSH |C|THING |C|RUBBER |C|ROUND |C|METAL |C|GREEN
|C|PLASTIC |C|VIOLET |C|PURPLE |C|PINK |C|TRANSLUCENT |C|LENGTH |C|RED
|C|REL-WEIGHT |C|3D-ROUND |C|2D-ROUND |C|CYLINDRICAL |C|REL-SIZE
|C|REL-LENGTH |C|DEFAULTVALUE )C|LISPDATA |C|BLUE |C|SIZE |C|THICK
|C|CONCEPT |C|THIN |C|ICONCEPT  |C|ROLE |C|UNKNOWN |C|BLACK |C|IROLE
|C|2D-END |C|CYLINDER-DIMENSIONS |C|TUBE |C|FUNCTIONAL-OBJECT |C|3D-END
|C|PARALLELEPIPED |C|BRICK |C|BRICK-DIMENSIONS |C|CYLINDER |C|INCHES
|C|SIDE |C|MEASURE-UNIT |C|DEGREES |C|3D-TRANSLATION |C|3D-ROTATION
|C|MOVEMENT |C|ATTACHING-DEVICE |C|THREADS |C|BOTTOM |C|THREADED-END
|C|TOP |C|VALVE |C|2D-OBJECT |C|UNTHREADED-END |C|OPENING |C|HOLE
|C|REL-POSITION NIL

62 <- (* Now need to create a concept that represents the speaker's
        description.  Here is a possible result of semantic
        interpretation of the noun phrase "a violet metal tube with a
        cylinderical outlet."  The function RC is used to create
        a real KL-One concept described in a notational form called
        NT.)

64 <- (RC (concept TESTTUBE (specializes TUBE)
            (roleset COLOR (mods (roleset COLOR of TUBE))
                      (vd (concept VIOLET (specializes COLOR))))
            (roleset COMPOSITION (mods (roleset COMPOSITION of TUBE))
                      (vd (concept METAL (specializes MATERIAL))))
            (roleset OUTLET (diffs (roleset SUBPART of PHYSICAL-OBJECT))
                      (vd (concept CYLINDER#99 (specializes CYLINDER]
```

FIGURE E-1, CONTINUED

```
(|C|TESTTUBE)

65 <- PPC TESTTUBE T

TESTTUBE
  type: Generic
  specializes: TUBE, PHYSICAL-OBJECT
  roles:
     COLOR
       Mods (COLOR of TUBE)
       V/R = VIOLET
     COMPOSITION
       Mods (COMPOSITION of TUBE)
       V/R = METAL
     OUTLET
       Diffs (SUBPART of PHYSICAL-OBJECT)
       V/R = CYLINDER#99

66 <- (* Now will put TESTTUBE into the linguistic focus, FOCUS39A.)

67 <- (RC (concept TESTTUBE (specializes FOCUS39A]

(|C|TESTTUBE)

68 <- PPC TESTTUBE T

TESTTUBE
  type: Generic
  specializes: TUBE, PHYSICAL-OBJECT, FOCUS39A
  roles:
     COLOR
       Mods (COLOR of TUBE)
       V/R = VIOLET
     COMPOSITION
       Mods (COMPOSITION of TUBE)
       V/R = METAL
     OUTLET
       Diffs (SUBPART of PHYSICAL-OBJECT)
       V/R = CYLINDER#99

70 <- (* Now classify TESTTUBE to see if i' fits in the current linguistic
 focus space, CONTEXT39.]

71 <- (Classify (KLGetNamedConcept 'TESTTUBE]

(|C|TESTTUBE ( |Cable|((** CableConnector 3) from |C|TESTTUBE to |C|TUBE4)
                |Cable|((** CableConnector 4) from |C|TESTTUBE to |C|TUBE1))
              NIL)

73 <- (* Since a SuperC cable was installed between TESTTUBE and TUBE1,
         it shows that the description TESTTUBE contains MORE
         information than was previous specified (from previous
         utterances) to describe  the current focus, FOCUS39A.
         TESTTUBE, hence, may not be an anaphoric expression that
```

FIGURE E-1, CONTINUED

```
             refers to the current real world element in focus (which can
             be found by looking at the RealWorld pointer on FOCUS39A).  If
             the SuperC cable had istead gone from TUBE1 to TESTTUBE,
             then TESTTUBE would have been considered to be an anaphoric
             referent to the same real world object as TUBE1.  Since it
             didn't happen this way, than it likely refers to something
             else.)

    74 <- PPC FOCUS39A

    FOCUS39A
      type: Generic [•]
      specializes: FOCUS, THING
      has specializers: TUBE1, TESTTUBE
      roles:
         (SubFocus of FOCUS)
           ValDescs = (FOCUS)
           Number = 1
           Modality = Optional
         SubFocus39A ((SubFocus of FOCUS))
           ValDescs = (SUBFOCUS39A)
           Number = 1
           Modality = Optional
      is role value of (Focus1 of CONTEXT39)
      has attached data:
         RealWorld       |C|MAIN-TUBE
         InTaxonomyFlg   T

    75 <- (• The RealWorld pointer on FOCUS39A shows the current real world
             element in focus is MAIN-TUBE.  Let's explore the taxonomy for
             other possible referents to see if focus could have shifted.
             Since the taxonomy exploration algorithm currently does NOT
             check to see that the elements are all in the same basic
             category - e.g., all of them are TUBEs, or PHYSICAL-OBJECTS -
             then it will end up collecting many elements that are clearly
             not reasonable referents.  These could easily be pruned off
             during the taxonomy exploration by having a BasicCategory
             datum attached to each concept.  When we explore a part of
             the taxonomy where the BasicCategory differs from the original,
             then we can prune off that part of the taxonomy and not
             bother exploring it any further.  Another way to prune off
             these unlikely elements is by letting the partial matcher
             score them very low.  I followed the latter route to save
             my time during the implementation but the former route is the
             best one.)

    76 <- (KLExploreTaxonomy (KLGetNamedConcept 'TESTTUBE]

    (|C|CYLINDER#1  |C|CYLINDER#2  |C|CYLINDER#3  |C|CYLINDER#4  |C|CYLINDER#5
                    |C|CYLINDER#6
                    |C|MAIN-TUBE  |C|CAP#1  |C|PLUNGER#1  |C|VALVE#1  |C|BRICK#1)

    93 <- (• The above are possible candidates for a referent for TESTTUBE.]
             Will demonstrate the KL-One partial matcher in action by
             comparing TESTTUBE to MAIN-TUBE.)
```

FIGURE E-1, CONTINUED

```
95 <- (KLPartialMatch (KLGetNamedConcept 'TESTTUBE)
                      (KLGetNamedConcept 'MAIN-TUBE]

((|R|(ORIENTATION OF PHYSICAL-OBJECT) |R|(ORIENTATION OF PHYSICAL-OBJECT) 27)
 (|R|(THICKNESS OF PHYSICAL-OBJECT) |R|(THICKNESS OF PHYSICAL-OBJECT) 27)
 (|R|(SUBPART OF PHYSICAL-OBJECT) NIL NIL)
 (|R|(POSITION OF PHYSICAL-OBJECT) |R|(POSITION OF PHYSICAL-OBJECT) 27)
 (|R|(COLOR OF TESTTUBE) |R|(COLOR OF MAIN-TUBE) 30)
 (|R|(COMPOSITION OF CYLINDER) |R|(COMPOSITION OF CYLINDER) 16.2)
 (|R|(COMPOSITION OF PHYSICAL-OBJECT) |R|(COMPOSITION OF PHYSICAL-OBJECT) 16.2)
 (|R|(COMPOSITION OF TESTTUBE) |R|(COMPOSITION OF MAIN-TUBE) 24)
 (|R|(OUTLET OF TESTTUBE) |R|(TUBE OF MAIN-TUBE) 20)
 (|R|(TRANSPARENCY OF PHYSICAL-OBJECT) |R|(TRANSPARENCY OF PHYSICAL-OBJECT)
                                       16.2)
 (|R|(ORIENTATION OF CYLINDER) |R|(ORIENTATION OF CYLINDER) 16.2)
 (|R|(STRENGTH OF PHYSICAL-OBJECT) |R|(STRENGTH OF PHYSICAL-OBJECT) 27)
 (|R|(MATTER OF PHYSICAL-OBJECT) |R|(MATTER OF PHYSICAL-OBJECT) 27)
 (|R|(TRANSPARENCY OF TUBE) |R|(TRANSPARENCY OF MAIN-TUBE) 15.0)
 (|R|(SUBFOCUS39A OF TESTTUBE) |R|(THREADS OF MAIN-TUBE) 10)
 (|R|(SUBFOCUS OF FOCUS) NIL NIL)
 (|R|(END OF TESTTUBE) |R|(LIP OF MAIN-TUBE) 23)
 (|R|(DIMENSIONS OF TESTTUBE) |R|(DIMENSIONS OF CYLINDER) 27)
 (|R|(WEIGHT OF PHYSICAL-OBJECT) |R|(WEIGHT OF PHYSICAL-OBJECT) 27))


102 <- (* The above show how the roles on each concept were aligned
          to each other, and what score was generated for that role
          alignment.)

114 <- (* Now show the complete alignment of TESTTUBE and MAIN-TUBE.
          By complete alignment, I mean show ALL possible role
          alignments and not just the best one.)

115 <- (KLAlignConcepts (KLGetNamedConcept 'TESTTUBE)
                        (KLGetNamedConcept 'MAIN-TUBE]

((|R|(WEIGHT OF PHYSICAL-OBJECT) ((|R|(WEIGHT OF PHYSICAL-OBJECT) (> + + + + 27))))
 (|R|(DIMENSIONS OF TESTTUBE) ((|R|(DIMENSIONS OF CYLINDER) (> + + + + 27))))
 (|R|(END OF TESTTUBE) ((|R|(LIP OF MAIN-TUBE) (+ ? > + + 23))
                        (|R|(THREADS OF MAIN-TUBE) (+ ? > + + 23))
                        (|R|(TUBE OF MAIN-TUBE) (+ ? = + + 20))
                        (|R|(OUTLET2 OF MAIN-TUBE) (+ ? = + + 20))
                        (|R|(OUTLET1 OF MAIN-TUBE) (+ ? = + + 20))))
 (|R|(SUBFOCUS39A OF TESTTUBE) ((|R|(DIMENSIONS OF CYLINDER) (- ? - + + 10))
                                (|R|(LIP OF MAIN-TUBE) (- ? - + + 10))
                                (|R|(TUBE OF MAIN-TUBE) (- ? - + + 10))
                                (|R|(THREADS OF MAIN-TUBE) (- ? - + + 10))
                                (|R|(COLOR OF MAIN-TUBE) (- ? - + + 10))
                                (|R|(OUTLET1 OF MAIN-TUBE) (- ? - + + 10))
                                (|R|(OUTLET2 OF MAIN-TUBE) (- ? - + + 10))
                                (|R|(POSITION OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(COMPOSITION OF MAIN-TUBE) (- ? - + + 10))
                                (|R|(STRENGTH OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(ORIENTATION OF PHYSICAL-OBJECT) (- ? - + + 10))
```

FIGURE E-1, CONTINUED

```
                                (|R|(WEIGHT OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(TRANSPARENCY OF MAIN-TUBE) (- ? - + + 10))
                                (|R|(MATTER OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(THICKNESS OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(COMPOSITION OF CYLINDER) (- < - + + 7.2))
                                (|R|(ORIENTATION OF CYLINDER) (- < - + + 7.2))
                                (|R|(COMPOSITION OF PHYSICAL-OBJECT) (- ? - + + 6.0))
                                (|R|(TRANSPARENCY OF PHYSICAL-OBJECT) (- ? - + + 6.0))))
 (|R|(TRANSPARENCY OF TUBE) ((|R|(TRANSPARENCY OF MAIN-TUBE) (> > + + + 15.0))))
 (|R|(MATTER OF PHYSICAL-OBJECT) ((|R|(MATTER OF PHYSICAL-OBJECT) (> + + + + 27))))
 (|R|(STRENGTH OF PHYSICAL-OBJECT) ((|R|(STRENGTH OF PHYSICAL-OBJECT) (> + + + +27))))
 (|R|(ORIENTATION OF CYLINDER) ((|R|(ORIENTATION OF CYLINDER) (> + + + + 16.2))))
 (|R|(TRANSPARENCY OF PHYSICAL-OBJECT) ((|R|(TRANSPARENCY OF PHYSICAL-OBJECT)
                                            (> + + + + 16.2))))
 (|R|(OUTLET OF TESTTUBE) ((|R|(LIP OF MAIN-TUBE) (+ ? = + + 20))
                           (|R|(TUBE OF MAIN-TUBE) (+ ? = + + 20))
                           (|R|(OUTLET2 OF MAIN-TUBE) (+ ? = + + 20))
                           (|R|(THREADS OF MAIN-TUBE) (+ ? = + + 20))
                           (|R|(OUTLET1 OF MAIN-TUBE) (+ ? = + + 20))))
 (|R|(COMPOSITION OF TESTTUBE) ((|R|(COMPOSITION OF MAIN-TUBE) (+ ? + + + 24))))
 (|R|(COMPOSITION OF PHYSICAL-OBJECT) ((|R|(COMPOSITION OF PHYSICAL-OBJECT)
                                         (> + + + + 16.2))))
 (|R|(COMPOSITION OF CYLINDER) ((|R|(COMPOSITION OF CYLINDER) (> + + + + 16.2))))
 (|R|(COLOR OF TESTTUBE) ((|R|(COLOR OF MAIN-TUBE) (+ + + + + 30))))
 (|R|(POSITION OF PHYSICAL-OBJECT) ((|R|(POSITION OF PHYSICAL-OBJECT) (> + + + + 27))))
 (|R|(THICKNESS OF PHYSICAL-OBJECT) ((|R|(THICKNESS OF PHYSICAL-OBJECT)
                                       (> + + + + 27))))
 (|R|(ORIENTATION OF PHYSICAL-OBJECT) ((|R|(ORIENTATION OF PHYSICAL-OBJECT)
                                         (> + + + + 27)))))


120 <-   (* Score the matches between TESTTUBE and MAINTUBE.]

123 <- (KLScoreConcept (KLEvaluateRoleMatch (VALUEOF 115]

(41 42)   (* The score has two parts.  One is a MINIMUM score
             and the other is a MAXIMUM score.  This is necessary
             because locally maximizing role alignment scores
             won't necessarily result in a maximum concept score.
             Hence I return a range of concept scores to be more
             accurate.)

125 <- (* We will try scoring TESTTUBE against other things (i.e., other
          possible candidates).  Since the plunger, PLUNGER#1, is very
          different in many ways but has a similar shape, we will try
          it.)

126 <- (KLAlignConcepts (KLGetNamedConcept 'TESTTUBE)
                        (KLGetNamedConcept 'PLUNGER#1]

((|R|(WEIGHT OF PHYSICAL-OBJECT) ((|R|(WEIGHT OF PHYSICAL-OBJECT) (> + + + + 27))))
 (|R|(DIMENSIONS OF TESTTUBE) ((|R|(DIMENSIONS OF CYLINDER) (> + + + + 27))))
 (|R|(END OF TESTTUBE) ((|R|(END OF CYLINDER) (> ? + + + 21))))
 (|R|(SUBFOCUS39A OF TESTTUBE) ((|R|(DIMENSIONS OF CYLINDER) (- ? - + + 10))
                                (|R|(COMPOSITION OF PLUNGER#1) (- ? - + + 10))
```

FIGURE E-1, CONTINUED

```
                                  (|R|(POSITION OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(COLOR OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(TRANSPARENCY OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(STRENGTH OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(MATTER OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(END OF CYLINDER) (- ? - + + 10))
                                  (|R|(WEIGHT OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(ORIENTATION OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(THICKNESS OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(ORIENTATION OF CYLINDER) (- < - + + 7.2))
                                  (|R|(COMPOSITION OF PHYSICAL-OBJECT) (- ? - + + 6.0))))
  (|R|(TRANSPARENCY OF TUBE) ((|R|(TRANSPARENCY OF PHYSICAL-OBJECT) (? > + + + 13.8))
                               (|R|(ORIENTATION OF CYLINDER) (? + - + + 10.8))
                               (|R|(DIMENSIONS OF CYLINDER) (? > - + + 9.6))
                               (|R|(COMPOSITION OF PLUNGER#1) (? > - + + 9.6))
                               (|R|(ORIENTATION OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(POSITION OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(COMPOSITION OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(COLOR OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(WEIGHT OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(THICKNESS OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(STRENGTH OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(MATTER OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(END OF CYLINDER) (? > - + + 9.6))))
  (|R|(MATTER OF PHYSICAL-OBJECT) ((|R|(MATTER OF PHYSICAL-OBJECT) (> + + + + 27))))
  (|R|(STRENGTH OF PHYSICAL-OBJECT) ((|R|(STRENGTH OF PHYSICAL-OBJECT) (> + + + + 27))))
  (|R|(ORIENTATION OF CYLINDER) ((|R|(ORIENTATION OF CYLINDER) (> + + + + 16.2))))
  (|R|(TRANSPARENCY OF PHYSICAL-OBJECT) ((|R|(TRANSPARENCY OF PHYSICAL-OBJECT)
                                            (> + + + + 27))))
  (|R|(OUTLET OF TESTTUBE) ((|R|(END OF CYLINDER) (> ? = + + 17))))
  (|R|(COMPOSITION OF TESTTUBE) ((|R|(COMPOSITION OF PLUNGER#1) (- + + + + 23))
                                  (|R|(COMPOSITION OF PHYSICAL-OBJECT) (- < + + + 11.4))
                                  (|R|(DIMENSIONS OF CYLINDER) (- ? - + + 10))
                                  (|R|(POSITION OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(TRANSPARENCY OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(COLOR OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(STRENGTH OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(MATTER OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(END OF CYLINDER) (- ? - + + 10))
                                  (|R|(WEIGHT OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(ORIENTATION OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(THICKNESS OF PHYSICAL-OBJECT) (- ? - + + 10))
                                  (|R|(ORIENTATION OF CYLINDER) (- < - + + 7.2))))
  (|R|(COMPOSITION OF PHYSICAL-OBJECT) ((|R|(COMPOSITION OF PHYSICAL-OBJECT)
                                             (> + + + + 16.2))))
  (|R|(COMPOSITION OF CYLINDER) ((|R|(COMPOSITION OF PLUNGER#1) (> > + + + 15.0))))
  (|R|(COLOR OF TESTTUBE) ((|R|(COLOR OF PHYSICAL-OBJECT) (> < + + + 23))))
  (|R|(POSITION OF PHYSICAL-OBJECT) ((|R|(POSITION OF PHYSICAL-OBJECT) (> + + + + 27))))
  (|R|(THICKNESS OF PHYSICAL-OBJECT) ((|R|(THICKNESS OF PHYSICAL-OBJECT)
                                          (> + + + + 27))))
  (|R|(ORIENTATION OF PHYSICAL-OBJECT) ((|R|(ORIENTATION OF PHYSICAL-OBJECT)
                                          (> + + + + 27)))))

127 <- (KLScoreConcept (KLEvaluateRoleMatch IT]
```

FIGURE E-1, CONTINUED

```
(40 41)

128 <- PPC PLUNGER#1 T

PLUNGER#1
  type: Individual
  individuates: PLUNGER, REALWORLDCATCHALLFOCUS, CYLINDER
  roles:
     COMPOSITION
       Mods (COMPOSITION of CYLINDER)
       V/R = METAL
  has attached data:
     InTaxonomyFlg  T

130 <- PPC TESTTUBE T

TESTTUBE
  type: Generic
  specializes: TUBE, PHYSICAL-OBJECT, FOCUS39A, TUBE1, TUBE4,
               PARALLELEPIPED, Agent
  roles:
     COLOR
       Mods (COLOR of TUBE), (COLOR of TUBE1), (COLOR of TUBE4)
       V/R = VIOLET
     COMPOSITION
       Mods (COMPOSITION of TUBE)
       V/R = METAL
     OUTLET
       Diffs (SUBPART of PHYSICAL-OBJECT), (SUBPART of PHYSICAL-OBJECT),
             (SUBPART of PHYSICAL-OBJECT), (SUBPART of PHYSICAL-OBJECT)
       V/R = CYLINDER#99
     SubFocus39A
       Diffs (SubFocus of FOCUS), (SubFocus of FOCUS)
       Mods (SubFocus39A of FOCUS39A)
     END
       Diffs (SUBPART of PHYSICAL-OBJECT), (SUBPART of PHYSICAL-OBJECT),
             (SUBPART of PHYSICAL-OBJECT)
       Mods (END of TUBE), (END of PARALLELEPIPED)
     DIMENSIONS
       Mods (DIMENSIONS of CYLINDER)
  has attached data:
     InTaxonomyFlg  T

131 <- PPC BRICK#1 T

BRICK#1
  type: Individual
  individuates: BRICK
  roles:
     DIMENSIONS
       Mods (DIMENSIONS of BRICK)
       VAL = BRICK-DIMENSIONS#1
     ORIENTATION
       Mods (ORIENTATION of BRICK)
```

FIGURE E-1, CONTINUED

```
       VAL = ORIENTATION#1
  has attached data:
     InTaxonomyFlg  T

132 <- (* Try matching TESTTUBE to BRICK#1.)

133 <- (KLAlignConcepts (KLGetNamedConcept 'TESTTUBE)
                        (KLGetNamedConcept 'BRICK#1]

((|R|(WEIGHT OF PHYSICAL-OBJECT) ((|R|(WEIGHT OF PHYSICAL-OBJECT) (> + + + + 27))))
 (|R|(DIMENSIONS OF TESTTUBE) ((|R|(DIMENSIONS OF BRICK#1) (+ ? + + + 24))))
 (|R|(END OF TESTTUBE) ((|R|(END OF PARALLELEPIPED) (> ? + + + 21))))
 (|R|(SUBFOCUS39A OF TESTTUBE) ((|R|(ORIENTATION OF PHYSICAL-OBJECT) (> + + + + 16.2))
                                (|R|(ORIENTATION OF BRICK#1) (- ? - + + 10))
                                (|R|(END OF PARALLELEPIPED) (- ? - + + 10))
                                (|R|(THICKNESS OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(COLOR OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(POSITION OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(COMPOSITION OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(WEIGHT OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(TRANSPARENCY OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(STRENGTH OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(MATTER OF PHYSICAL-OBJECT) (- ? - + + 10))
                                (|R|(DIMENSIONS OF BRICK#1) (- ? - + + 10))
                                (|R|(COLOR OF BRICK) (- < - + + 7.2))
                                (|R|(COMPOSITION OF BRICK) (- < - + + 7.2))))
 (|R|(TRANSPARENCY OF TUBE) ((|R|(TRANSPARENCY OF PHYSICAL-OBJECT) (? > + + + 13.8))
                             (|R|(COLOR OF BRICK) (? + - + + 10.8))
                             (|R|(COMPOSITION OF BRICK) (? + - + + 10.8))
                             (|R|(ORIENTATION OF BRICK#1) (? > - + + 9.6))
                             (|R|(END OF PARALLELEPIPED) (? > - + + 9.6))
                             (|R|(THICKNESS OF PHYSICAL-OBJECT) (? > - + + 9.6))
                             (|R|(POSITION OF PHYSICAL-OBJECT) (? > - + + 9.6))
                             (|R|(ORIENTATION OF PHYSICAL-OBJECT) (? > - + + 9.6))
                             (|R|(COMPOSITION OF PHYSICAL-OBJECT) (? > - + + 9.6))
                             (|R|(COLOR OF PHYSICAL-OBJECT) (? > - + + 9.6))
                             (|R|(WEIGHT OF PHYSICAL-OBJECT) (? > - + + 9.6))
                             (|R|(STRENGTH OF PHYSICAL-OBJECT) (? > - + + 9.6))
                             (|R|(MATTER OF PHYSICAL-OBJECT) (? > - + + 9.6))
                             (|R|(DIMENSIONS OF BRICK#1) (? > - + + 9.6))))
 (|R|(MATTER OF PHYSICAL-OBJECT) ((|R|(MATTER OF PHYSICAL-OBJECT) (> + + + + 27))))
 (|R|(STRENGTH OF PHYSICAL-OBJECT) ((|R|(STRENGTH OF PHYSICAL-OBJECT) (> + + + + 27))))
 (|R|(ORIENTATION OF CYLINDER) ((|R|(ORIENTATION OF BRICK#1) (? > + + + 13.8))
                                (|R|(ORIENTATION OF PHYSICAL-OBJECT) (? > + + + 13.8))
                                (|R|(COLOR OF BRICK) (? + - + + 10.8))
                                (|R|(COMPOSITION OF BRICK) (? + - + + 10.8))
                                (|R|(END OF PARALLELEPIPED) (? > - + + 9.6))
                                (|R|(COLOR OF PHYSICAL-OBJECT) (? > - + + 9.6))
                                (|R|(THICKNESS OF PHYSICAL-OBJECT) (? > - + + 9.6))
                                (|R|(POSITION OF PHYSICAL-OBJECT) (? > - + + 9.6))
                                (|R|(COMPOSITION OF PHYSICAL-OBJECT) (? > - + + 9.6))
                                (|R|(WEIGHT OF PHYSICAL-OBJECT) (? > - + + 9.6))
                                (|R|(TRANSPARENCY OF PHYSICAL-OBJECT) (? > - + + 9.6))
                                (|R|(STRENGTH OF PHYSICAL-OBJECT) (? > - + + 9.6))
                                (|R|(MATTER OF PHYSICAL-OBJECT) (? > - + + 9.6))
```

FIGURE E-1, CONTINUED

```
                               (|R|(DIMENSIONS OF BRICK#1) (? > - + + 9.6))))
(|R|(TRANSPARENCY OF PHYSICAL-OBJECT) ((|R|(TRANSPARENCY OF PHYSICAL-OBJECT)
                                            (> + + + + 27))))
(|R|(OUTLET OF TESTTUBE) ((|R|(END OF PARALLELEPIPED) (> ? = + + 17))))
(|R|(COMPOSITION OF TESTTUBE) ((|R|(COMPOSITION OF PHYSICAL-OBJECT) (- < + + + 19))
                               (|R|(ORIENTATION OF PHYSICAL-OBJECT) (> + + + + 16.2))
                               (|R|(COMPOSITION OF BRICK) (- < + + + 11.4))
                               (|R|(ORIENTATION OF BRICK#1) (- ? - + + 10))
                               (|R|(END OF PARALLELEPIPED) (- ? - + + 10))
                               (|R|(THICKNESS OF PHYSICAL-OBJECT) (- ? - + + 10))
                               (|R|(POSITION OF PHYSICAL-OBJECT) (- ? - + + 10))
                               (|R|(WEIGHT OF PHYSICAL-OBJECT) (- ? - + + 10))
                               (|R|(COLOR OF PHYSICAL-OBJECT) (- ? - + + 10))
                               (|R|(TRANSPARENCY OF PHYSICAL-OBJECT) (- ? - + + 10))
                               (|R|(STRENGTH OF PHYSICAL-OBJECT) (- ? - + + 10))
                               (|R|(MATTER OF PHYSICAL-OBJECT) (- ? - + + 10))
                               (|R|(DIMENSIONS OF BRICK#1) (- ? - + + 10))
                               (|R|(COLOR OF BRICK) (- < - + + 7.2))))
(|R|(COMPOSITION OF PHYS:CAL-OBJECT) ((|R|(COMPOSITION OF PHYSICAL-OBJECT)
                                          (> + + + + 16.2))))
(|R|(COMPOSITION OF CYLINDER) ((|R|(COMPOSITION OF BRICK) (? + + + + 15.0))
                               (|R|(COMPOSITION OF PHYSICAL-OBJECT) (? > + + + 13.8))
                               (|R|(COLOR OF BRICK) (? + - + + 10.8))
                               (|R|(ORIENTATION OF BRICK#1) (? > - + + 9.6))
                               (|R|(END OF PARALLELEPIPED) (? > - + + 9.6))
                               (|R|(THICKNESS OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(POSITION OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(ORIENTATION OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(COLOR OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(WEIGHT OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(TRANSPARENCY OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(STRENGTH OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(MATTER OF PHYSICAL-OBJECT) (? > - + + 9.6))
                               (|R|(DIMENSIONS OF BRICK#1) (? > - + + 9.6))))
(|R|(COLOR OF TESTTUBE) ((|R|(COLOR OF PHYSICAL-OBJECT) (> < + + + 23))))
(|R|(POSITION OF PHYSICAL-OBJECT) ((|R|(POSITION OF PHYSICAL-OBJECT) (> + + +. + 27))))
                                   )
(|R|(THICKNESS OF PHYSICAL-OBJECT) ((|R|(THICKNESS OF PHYSICAL-OBJECT)
                                          (> + + + + 27))))
(|R|(ORIENTATION OF PHYSICAL-OBJECT) ((|R|(ORIENTATION OF PHYSICAL-OBJECT)
                                          (> + + + + 16.2)))))


134 <-  (KLScoreConcept (KLEvaluateRoleMatch IT]
(37 39)

(* The result of the partial matching and scoring is as follows:
      TESTTUBE to MAINTUBE:  (41 42);
      TESTTUBE to PLUNGER#1:  (40 41); and
      TESTTUBE to BRICK#1:  (37 39).
   Thus, MAINTUBE and PLUNGER#1 are the most likely referent
   candidates.  At this point, relaxation rules could be used
   to attempt to relax TESTTUBE to one of the candidates.)
```

FIGURE E-1, CONCLUDED

APPENDIX F
AN EXAMPLE FOR FINDING FEASIBLE REFERENT CANDIDATES

This appendix shows in Figure F-1 how the reference mechanism explores the taxonomy for referent candidates.

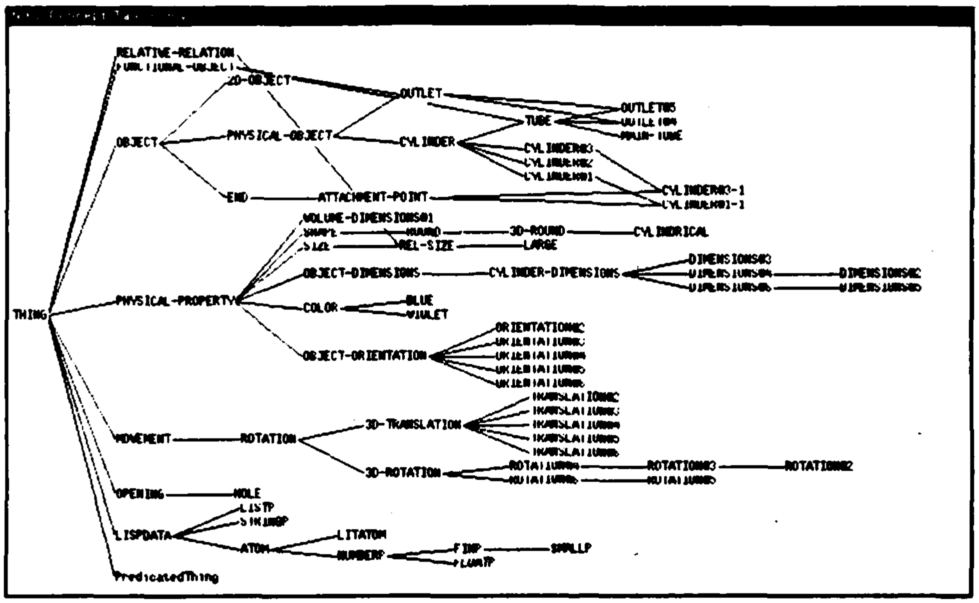
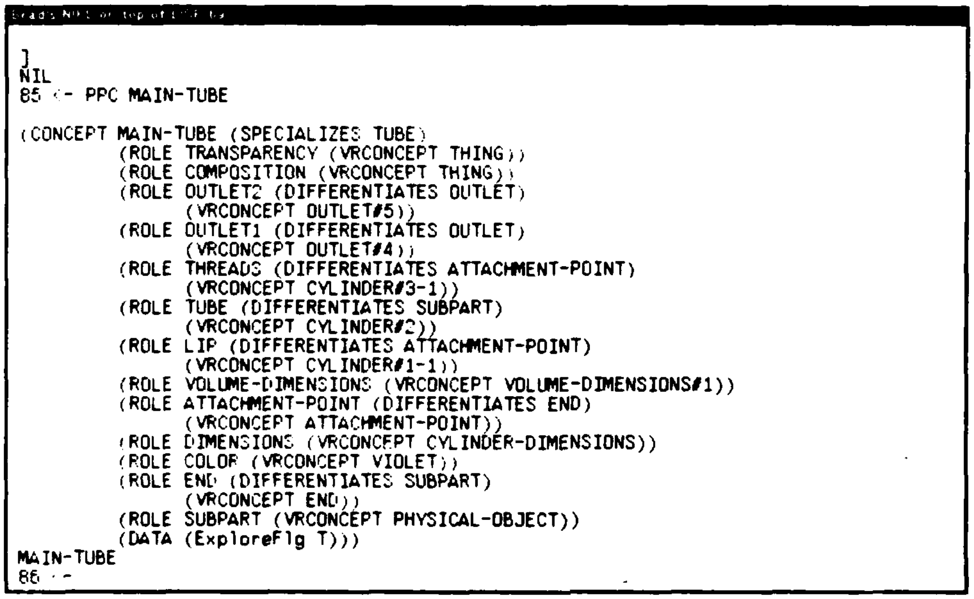
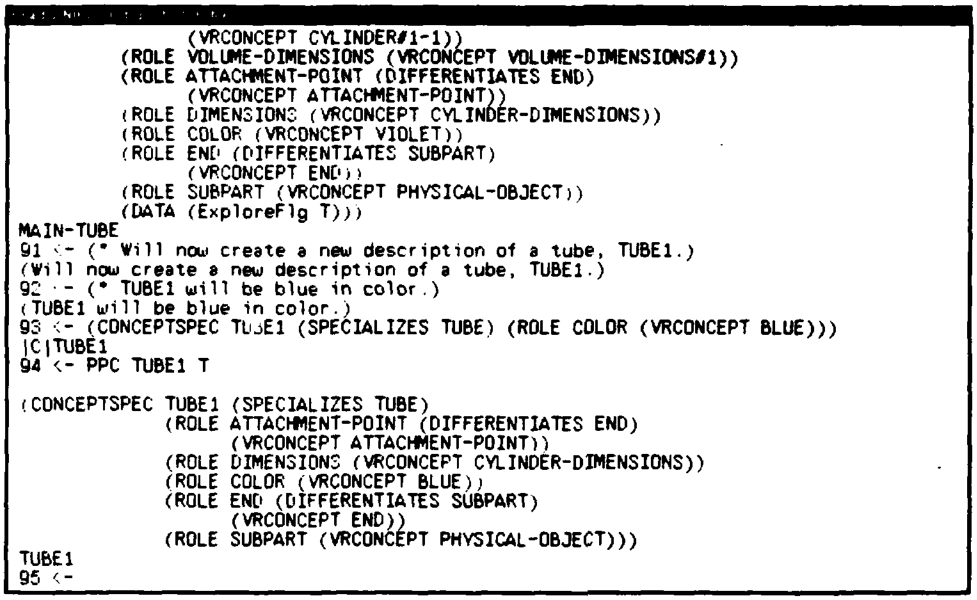


Figure F-1: Searching for referent candidates



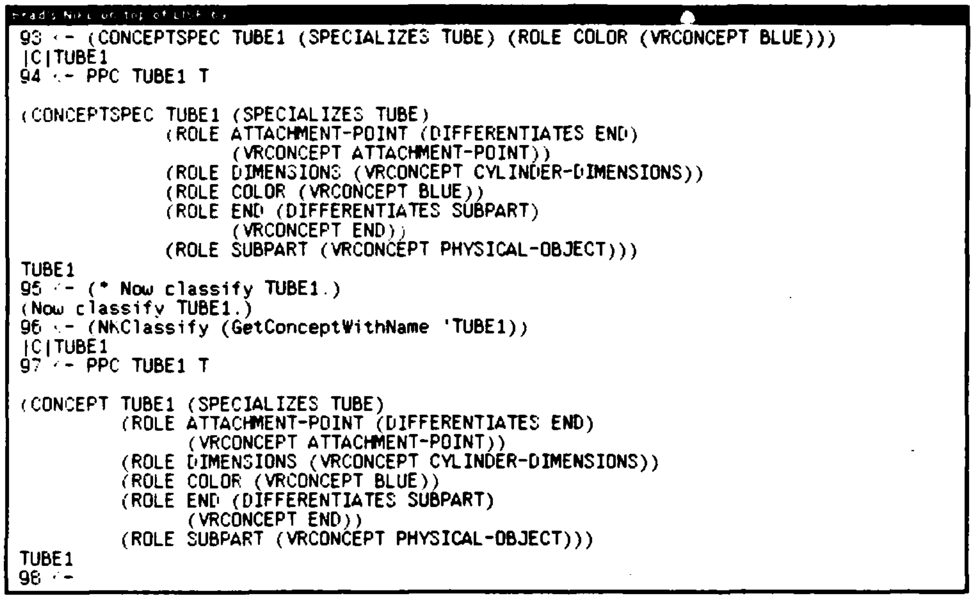
```
]
NIL
85 <- PPC MAIN-TUBE

(CONCEPT MAIN-TUBE (SPECIALIZES TUBE)
          (ROLE TRANSPARENCY (VRCONCEPT THING))
          (ROLE COMPOSITION (VRCONCEPT THING))
          (ROLE OUTLET2 (DIFFERENTIATES OUTLET)
                (VRCONCEPT OUTLET#5))
          (ROLE OUTLET1 (DIFFERENTIATES OUTLET)
                (VRCONCEPT OUTLET#4))
          (ROLE THREADS (DIFFERENTIATES ATTACHMENT-POINT)
                (VRCONCEPT CYLINDER#3-1))
          (ROLE TUBE (DIFFERENTIATES SUBPART)
                (VRCONCEPT CYLINDER#2))
          (ROLE LIP (DIFFERENTIATES ATTACHMENT-POINT)
                (VRCONCEPT CYLINDER#1-1))
          (ROLE VOLUME-DIMENSIONS (VRCONCEPT VOLUME-DIMENSIONS#1))
          (ROLE ATTACHMENT-POINT (DIFFERENTIATES END)
                (VRCONCEPT ATTACHMENT-POINT))
          (ROLE DIMENSIONS (VRCONCEPT CYLINDER-DIMENSIONS))
          (ROLE COLOR (VRCONCEPT VIOLET))
          (ROLE END (DIFFERENTIATES SUBPART)
                (VRCONCEPT END))
          (ROLE SUBPART (VRCONCEPT PHYSICAL-OBJECT))
          (DATA (ExploreFlg T)))
MAIN-TUBE
86 <-
```

FIGURE F-1, CONTINUED



```
                (VRCONCEPT CYLINDER#1-1))
         (ROLE VOLUME-DIMENSIONS (VRCONCEPT VOLUME-DIMENSIONS#1))
         (ROLE ATTACHMENT-POINT (DIFFERENTIATES END)
                (VRCONCEPT ATTACHMENT-POINT))
         (ROLE DIMENSIONS (VRCONCEPT CYLINDER-DIMENSIONS))
         (ROLE COLOR (VRCONCEPT VIOLET))
         (ROLE END (DIFFERENTIATES SUBPART)
                (VRCONCEPT END))
         (ROLE SUBPART (VRCONCEPT PHYSICAL-OBJECT))
         (DATA (ExploreFlg T)))
MAIN-TUBE
91 <- (* Will now create a new description of a tube, TUBE1.)
(Will now create a new description of a tube, TUBE1.)
92 <- (* TUBE1 will be blue in color.)
(TUBE1 will be blue in color.)
93 <- (CONCEPTSPEC TUBE1 (SPECIALIZES TUBE) (ROLE COLOR (VRCONCEPT BLUE)))
|C|TUBE1
94 <- PPC TUBE1 T

(CONCEPTSPEC TUBE1 (SPECIALIZES TUBE)
             (ROLE ATTACHMENT-POINT (DIFFERENTIATES END)
                    (VRCONCEPT ATTACHMENT-POINT))
             (ROLE DIMENSIONS (VRCONCEPT CYLINDER-DIMENSIONS))
             (ROLE COLOR (VRCONCEPT BLUE))
             (ROLE END (DIFFERENTIATES SUBPART)
                    (VRCONCEPT END))
             (ROLE SUBPART (VRCONCEPT PHYSICAL-OBJECT)))
TUBE1
95 <-
```

FIGURE F-1, CONTINUED



```
93 <- (CONCEPTSPEC TUBE1 (SPECIALIZES TUBE) (ROLE COLOR (VRCONCEPT BLUE)))
|C|TUBE1
94 <- PPC TUBE1 T

(CONCEPTSPEC TUBE1 (SPECIALIZES TUBE)
             (ROLE ATTACHMENT-POINT (DIFFERENTIATES END)
                   (VRCONCEPT ATTACHMENT-POINT))
             (ROLE DIMENSIONS (VRCONCEPT CYLINDER-DIMENSIONS))
             (ROLE COLOR (VRCONCEPT BLUE))
             (ROLE END (DIFFERENTIATES SUBPART)
                   (VRCONCEPT END))
             (ROLE SUBPART (VRCONCEPT PHYSICAL-OBJECT)))
TUBE1
95 <- (* Now classify TUBE1.)
(Now classify TUBE1.)
96 <- (NKClassify (GetConceptWithName 'TUBE1))
|C|TUBE1
97 <- PPC TUBE1 T

(CONCEPT TUBE1 (SPECIALIZES TUBE)
         (ROLE ATTACHMENT-POINT (DIFFERENTIATES END)
               (VRCONCEPT ATTACHMENT-POINT))
         (ROLE DIMENSIONS (VRCONCEPT CYLINDER-DIMENSIONS))
         (ROLE COLOR (VRCONCEPT BLUE))
         (ROLE END (DIFFERENTIATES SUBPART)
               (VRCONCEPT END))
         (ROLE SUBPART (VRCONCEPT PHYSICAL-OBJECT)))
TUBE1
98 <-
```

FIGURE F-1, CONTINUED

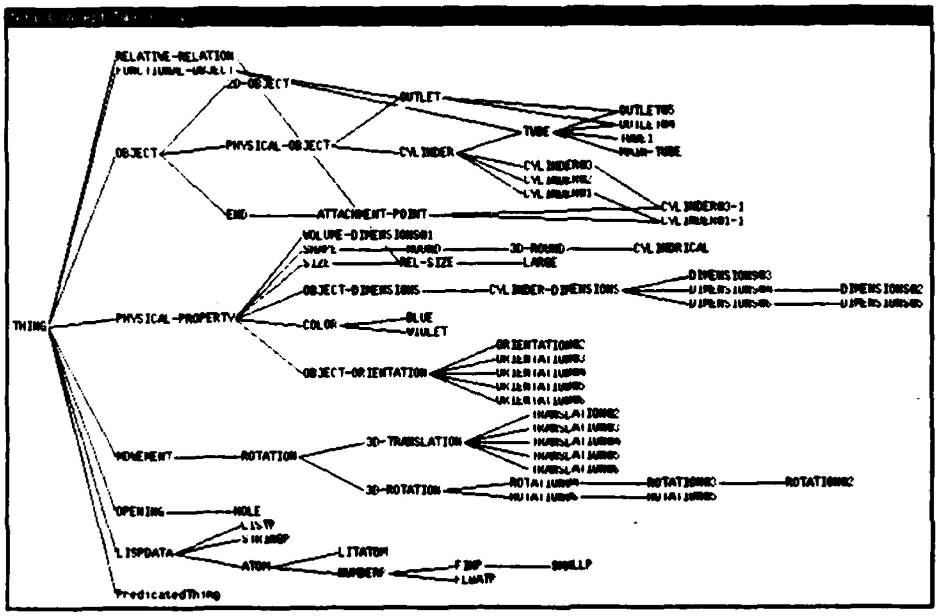
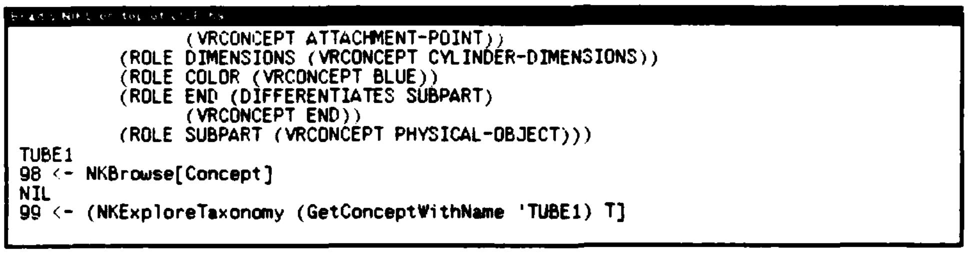


FIGURE F-1, CONTINUED



```
              (VRCONCEPT ATTACHMENT-POINT))
         (ROLE DIMENSIONS (VRCONCEPT CYLINDER-DIMENSIONS))
         (ROLE COLOR (VRCONCEPT BLUE))
         (ROLE END (DIFFERENTIATES SUBPART)
              (VRCONCEPT END))
         (ROLE SUBPART (VRCONCEPT PHYSICAL-OBJECT)))
TUBE1
98 <- NKBrowse[Concept]
NIL
99 <- (NKExploreTaxonomy (GetConceptWithName 'TUBE1) T]
```

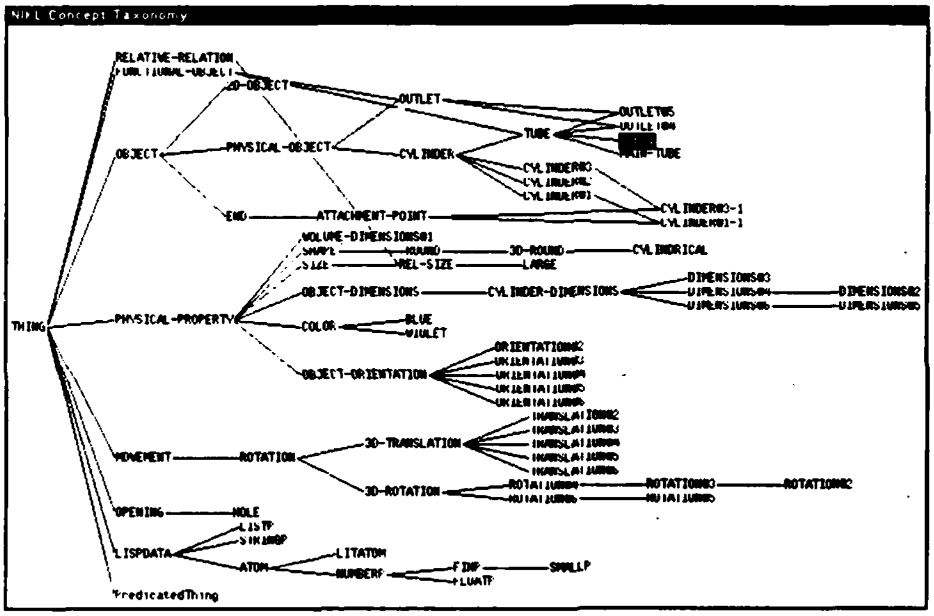


FIGURE F-1, CONTINUED

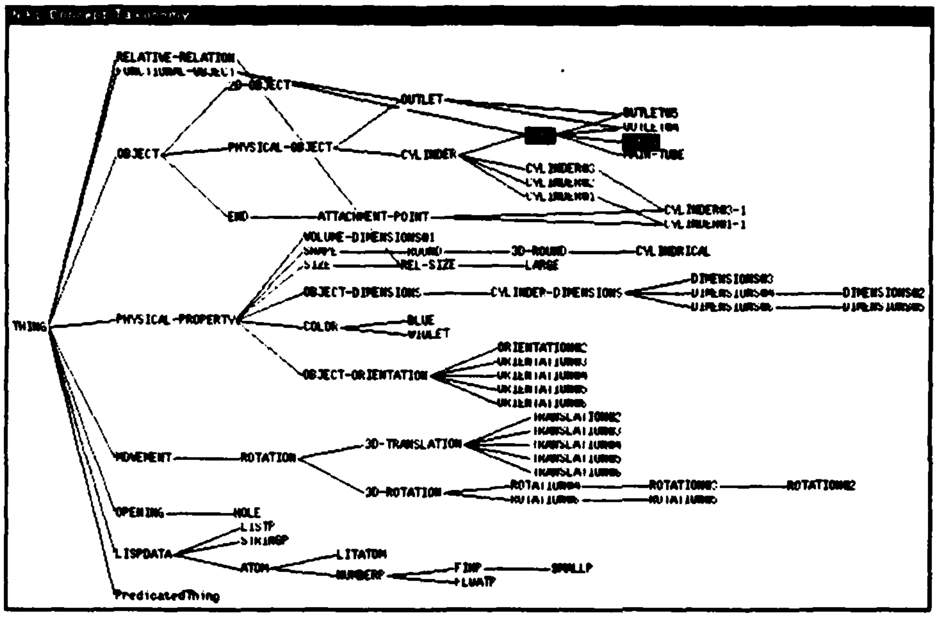


FIGURE F-1, CONTINUED

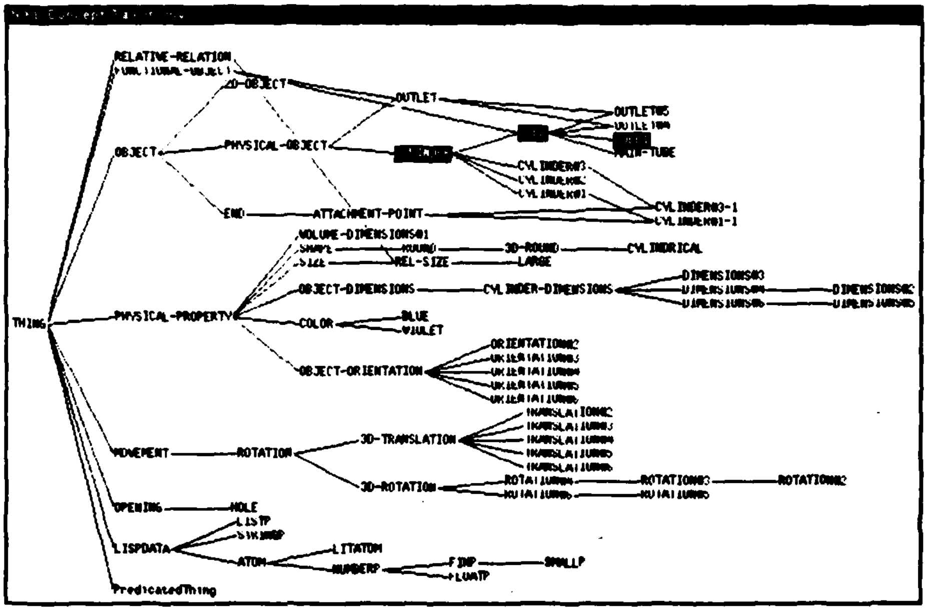


FIGURE F-1, CONTINUED

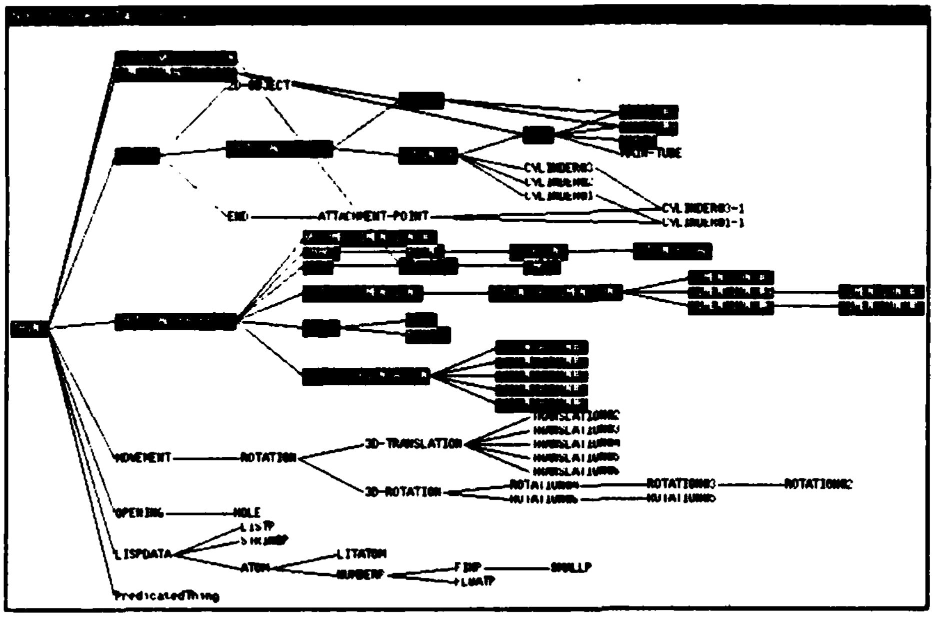


FIGURE F-1, CONTINUED

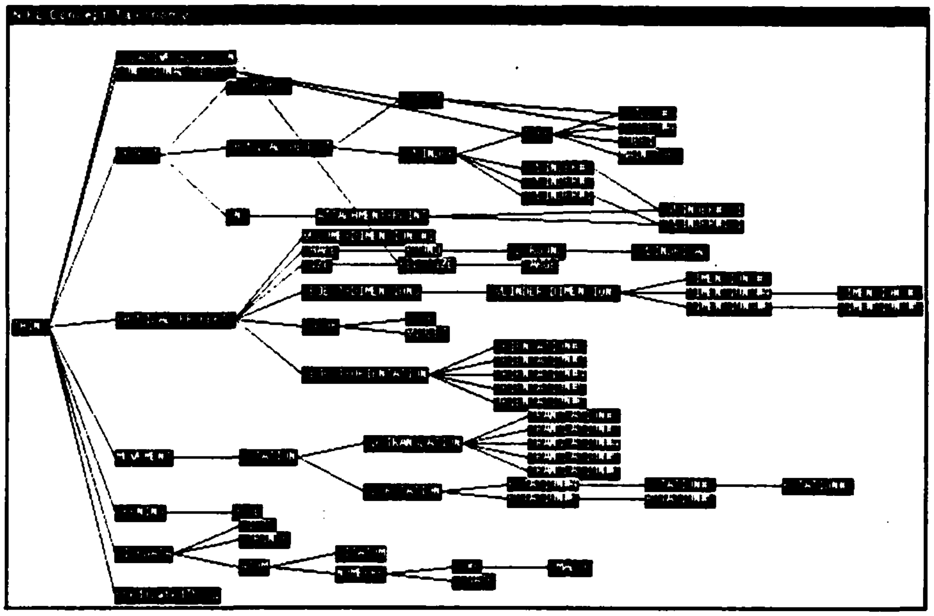


FIGURE F-1, CONCLUDED

References

[1] Agin, Gerald J. and Thomas O. Binford. "Computer Description of Curved Objects." *IEEE Transactions on Computers C-25*, 4 (April 1976), 439-449.

[2] Agin, Gerald J. Hierarchical Representation of Three-Dimensional Objects Using Verbal Models. Technical Note 182, SRI International, March, 1979.

[3] Allen, James F. *A Plan-Based Approach to Speech Act Recognition*. Ph.D. Th., University of Toronto, 1979.

[4] Allen, James F., Alan M. Frisch, and Diane J. Litman. ARGOT: The Rochester Dialogue System. Proceedings of AAAI-82, Pittsburgh, Pa., August, 1982, pp. 66-70.

[5] Appelt, Douglas E. *Planning Natural Language Utterances to Satisfy Multiple Goals*. Ph.D. Th., Stanford University, 1981.

[6] Bobrow, Robert J. The RUS System. BBN Report No. 3878, Bolt Beranek and Newman Inc., July, 1978.

[7] Bobrow, R. J. and B. L. Webber. PSI-KLONE - Parsing and Semantic Interpretation in the BBN Natural Language Understanding System. Proceedings Canadian Soc. for Computational Studies of Intelligence (CSCSI), Victoria, B.C., 1980.

[8] Brachman, Ronald J. *A Structural Paradigm for Representing Knowledge*. Ph.D. Th., Harvard University, 1977. Also, Technical Report No. 3605, Bolt Beranek and Newman Inc.

[9] Brachman, R. J., R. J. Bobrow, P. R. Cohen, J. W. Klovstad, B. L. Webber, W. A. Woods. Research in Natural Language Understanding, Annual Report: 1 Sept 78 - 31 Aug 79. Report No. 4274, Bolt Beranek and Newman Inc., August, 1979.

[10] Brachman, Ronald J. and James G. Schmolze. "An Overview of the KL-ONE Knowledge Representation System." *Cognitive Science 9*, 2 (1985), 171-216.

[11] Brown, John Seely and Kurt VanLehn. "Repair Theory: A Generative Theory of Bugs in Procedural Skills." *Cognitive Science 4*, 4 (1980), 379-426.

[12] Burling, R.. *Man's many voices: Language in its cultural context*. Holt; Rinehart and Winston, 1970.

[13] Burton, Richard R. Semantic Grammar: An Engineering Technique for Constructing Natural Language Understanding Systems. BBN Report No. 3453, Bolt Beranek and Newman Inc., December, 1976.

[14] Carbonell, Jaime R. and Allan M. Collins. Natural Semantics in Artificial Intelligence. Proceedings of IJCAI-73, Stanford, California, August, 1973, pp. 344-351.

[15] Charniak, Eugene. *Toward a Model of Children's Story Comprehension*. Ph.D. Th., Massachusetts Institute of Technology, 1972.

[16] Clark, Herbert H. and Eve V. Clark. *PSYCHOLOGY and LANGUAGE*. Harcourt Brace Jovanovich, Inc., 1977.

[17] Clark, H. H. and C. Marshall. Definite reference and mutual knowledge. In *Elements of Discourse Understanding*, Joshi, Webber and Sags, Ed.,Cambridge University Press, 1981, pp. 10-64.

[18] Cohen, Philip R. *On Knowing What to Say: Planning Speech Acts*. Ph.D. Th., University of Toronto, 1978.

[19] Cohen, P., C. Perrault and J. Allen. Beyond Question Answering. In *Knowledge Representation and Natural Language Processing*, W. Lehnart and M. Ringle, Ed.,Lawrence Erlbaum Associates, 1981.

[20] Cohen, Philip R. The need for Referent Identification as a Planned Action. Proceedings of IJCAI-81, Vancouver, B.C., Canada, August, 1981, pp. 31-35.

[21] Cohen, Philip R., Scott Fertig and Kathy Starr. Dependencies of Discourse Structure on the Modality of Communication: Telephone vs. Teletype. Proceedings of ACL, Toronto, Ont., Canada, June, 1982, pp. 28-35.

[22] Cohen, Philip R. "The Pragmatics of Referring and the Modality of Communication." *Computational Linguistics 10*, 2 (April-June 1984), 97-146.

[23] Fikes, Richard E. and Gary G. Hendrix. The Deduction Component. In *Understanding Spoken Language*, Donald E. Walker, Ed.,North-Holland, New York, 1978, pp. 355-374.

[24] Gentner, Dedre. The Structure of Analogical Models in Science. Bolt Beranek and Newman Inc., July, 1980.

[25] Goodman, Bradley A. A Model for a Natural Language Data Base System. Report R-798, Coordinated Science Laboratory, University of Illinois, October, 1977.

[26] Goodman, Bradley A. The Representation of Three-Dimensional Objects. KRNL Group Working Paper, Bolt Beranek and Newman Inc., December 1981.

[27] Goodman, Bradley A. Miscommunication in Task-Oriented Dialogues. KRNL Group Working Paper, Bolt Beranek and Newman Inc., April 1982.

[28] Goodman, Bradley A. Repairing Miscommunication: Relaxation in Reference. Proceedings of AAAI-83, Washington, D.C., August, 1983, pp. 134-138.

[29] Grice, H. P. Logic and Conversation: Implicature. In *Syntax and Semantics*, Cole and Morgan, Ed.,Academic Press, New York, 1975.

[30] Grosz, Barbara J. *The Representation and Use of Focus in Dialogue Understanding*. Ph.D. Th., University of California, Berkeley, 1977. Also, Technical Note 151, Stanford Research Institute.

[31] Grosz, Barbara J. Focusing in Dialog. Theoretical Issues in Natural Language Processing-2, Urbana, Ill., July, 1978, pp. 96-103.

[32] Grosz, Barbara J. Focusing and descriptions in natural language dialogues. In *Elements of Discourse Understanding*, Joshi, Webber and Sags, Ed.,Cambridge University Press, 1981, pp. 84-105.

[33] Halliday, M. A. K. "Functional Diversity in Language as Seen from a Consideration of Modality and Mood in English." *Foundations of Language 6* (1970), 322-361.

[34] Hendrix, Gary G. *Partitioned networks for the mathematical modeling of natural language semantics*. Ph.D. Th., University of Texas, Austin, 1975. Technical Report NL-28.

[35] Hendrix, Gary G. Semantic Knowledge. In *Understanding Spoken Language*, Donald E. Walker, Ed.,North-Holland, New York, 1978, pp. 121-226.

[36] Hewitt, Carl. PLANNER. Report No. MAC-M-386, Massachusetts Institute of Technology, October, 1968. Project MAC. Revised, August, 1970.

[37] Hoeppner, W., T. Christaller, H. Marburger, K. Morik, B. Nebel, M. O'Leary and W. Wahlster. BEYOND DOMAIN-INDEPENDENCE: Exp:rience with the Development of a German Language Access System To Highly Diverse Background Systems. Proceedings of IJCAI-83, Karlsruhe, West Germany, August, 1983, pp. 588-594.

[38] Joshi, Aravind K. Mutual Beliefs in Question-Answer Systems. In *Mutual Beliefs*, N. Smith, Ed.,Academic Press, 1982, pp. 181-197.

[39] Lipkis, Thomas. A KL-ONE Classifier. Proceedings of the 1981 KL-One Workshop, June, 1982, pp. 128-145. Report No. 4842, Bolt Beranek and Newman Inc. Also Consul Note # 5, USC/Information Sciences Institute, October 1981.

[40] Litman, Diane. Discourse and Problem Solving. Report No. 5338, Bolt Beranek and Newman Inc., July, 1983. Also, TR130, University of Rochester, Dept. of Computer Science.

[41] Litman, Diane J. and James F. Allen. A Plan Recognition Model for Clarification Subdialogues. Proceedings of Coling84, Stanford University, Stanford, CA., July, 1984, pp. 302-311.

[42] Litman, Diane J. *Plan Recognition and Discourse Analysis: An Integrated Approach for Understanding Dialogues*. Ph.D. Th., University of Rochester, 1985. Also, TR170, University of Rochester, Dept. of Computer Science.

[43] Mark, William. Realization. Proceedings of the 1981 KL-One Workshop, June, 1982, pp. 78-89. Report No. 4842, Bolt Beranek and Newman Inc.

[44] Marr, D. and H. K. Nishihara. Spatial disposition of axes in a generalized cylinder representation of objects that do not encompass the viewer. Memo No. 341, M.I.T. A.I. Lab, December, 1975.

[45] Marr, D. and H. K. Nishihara. Representation and recognition of the spatial organization of three dimensional shapes. Memo No. 416, M.I.T. A.I. Lab, May, 1977.

[46] Marr, David and H. Keith Nishihara. "Visual Information Processing: Artificial Intelligence and the Sensorium of Sight." *Technology Review* (October 1978), 28-49.

[47] McCoy, Kathleen F. The Role of Perspective in Responding to Property Misconceptions. Proceedings of IJCAI-85, Los Angeles, August, 1985, pp. 791-793.

[48] McDonald, David D. and E. Jeffery Conklin. Salience as a Simplifying Metaphor for Natural Language Generation. Proceedings of AAAI-82, Pittsburgh, Pa., August, 1982, pp. 75-78.

[49] McKeown, Kathleen R. Recursion in Text and Its Use in Language Generation. Proceedings of AAAI-83, Washington, D.C., August, 1983, pp. 270-273.

[50] Nadathur, Gopalan and Aravind K. Joshi. Mutual Beliefs in Conversational Systems: Their Role in Referring Expressions. Proceedings of IJCAI-83, Karlsruhe, West Germany, August, 1983, pp. 603-605.

[51] Norman, Donald A. and David E. Rumelhart. Reference and Comprehension. In *Explorations in Cognition*, D. A. Norman and D. E. Rumelhart, Ed.,W. H. Freeman and Company, 1975, pp. 65-87.

[52] Ochsman, Robert B. and Alphonse Chapanis. "The Effects of 10 Communication Modes on the Behavior of Teams During Cooperative Problem-solving." *Int. J. Man-Machine Studies 3* (1974), 579-619.

[53] Olson D. "Language and thought: Aspects of a cognitive theory of semantics." *Psychological Review 77*, 4 (1970), 257-273.

[54] Paxton, William H. The Language Definition System. In *Understanding Spoken Language*, Donald E. Walker, Ed.,North-Holland, New York, 1978, pp. 17-40.

[55] Perrault, C. Raymond and Philip R. Cohen. It's for your own good: a note on inaccurate reference. In *Elements of Discourse Understanding*, Joshi, Webber and Sags, Ed.,Cambridge University Press, 1981, pp. 217-230.

[56] Polanyi, Livia and Remko Scha. A Syntactic Approach to Discourse Semantics. Proceedings of Coling84, Stanford University, Stanford, CA., July, 1984, pp. 413-419.

[57] Reichman, Rachel. "Conversational Coherency." *Cognitive Science 2*, 4 (1978), 283-327.

[58] Reichman, Rachel. *Plain Speaking: A Theory and Grammar of Spontaneous Discourse*. Ph.D. Th., Harvard University, 1981. Also, Technical Report No. 4861, Bolt Beranek and Newman Inc.

[59] Rieger, Charles J. *Conceptual Memory: A Theory and Computer Program for Processing the Meaning of Natural Language Utterances.* Ph.D. Th., Stanford University, 1974.

[60] Ringle, Martin and Bertram Bruce. Conversation Failure. In *Knowledge Representation and Natural Language Processing*, W. Lehnart and M. Ringle, Ed.,Lawrence Erlbaum Associates, 1981.

[61] Robinson, A.E., Appelt, D.E., Grosz, B.J., Hendrix, G.G., & Robinson, J.J. Interpreting natural-language utterances in dialogs about tasks. Technical Note 210, Artificial Intelligence Center, SRI International, March, 1980.

[62] Robinson, Jane J. "DIAGRAM: A Grammar for Dialogues." *Communications of the ACM 25*, 1 (January 1982), 27-46.

[63] Robinson, Ann E. "Determining Verb Phrase Referents in Dialogs." *American Journal of Computational Linguistics 7*, 1 (1981), 1-16.

[64] Rosch, E. "Cognitive representations of semantic categories." *Journal of Experimental Psychology: General 104* (1975), 192-233.

[65] Rubin, Andee. A Theoretical Taxonomy of the Differences between Oral and Written Language. In *Theoretical Issues in Reading Understanding*, Rand J. Spiro, Bertram C. Bruce, and William F. Brewer, Ed.,Lawrence Erlbaum Associates, 1980.

[66] Schmolze, James G. and Thomas A. Lipkis. Classification in the KL-ONE Knowledge Representation System. Proceedings of IJCAI-83, Karlsruhe, West Germany, August, 1983, pp. 330-332.

[67] John R. Searle. *Speech Acts.* Cambridge University Press, 1969.

[68] Sidner, C. L., and Israel, D.J. Recognizing intended meaning and speaker's plans. Proceedings of the International Joint Conference in Artificial Intelligence, The International Joint Conferences on Artifical Intelligence, Vancouver, B.C., August, 1981, pp. 203-208.

[69] Sidner, Candace Lee. *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse.* Ph.D. Th., Massachusetts Institute of Technology, 1979. Also, Report No. TR-537, MIT AI Lab.

[70] Sidner, Candace L. Protocols of Users Manipulating Visually Presented Information for Natural Language. Report No. 5128, Bolt Beranek and Newman Inc., September, 1982.

[71] Sidner, C. L., M. Bates, R. J. Bobrow, R. J. Brachman, P. R. Cohen, D. J. Israel, J. Schmolze, B. L. Webber, W. A. Woods. Research in Knowledge Representation for Natural Language Understanding. Report No. 4785, Bolt Beranek and Newman Inc., November, 1981.

[72] Sidner, C. L., Bates, M., Bobrow, R., Goodman, B., Haas, A., Ingria, R., Israel, D., McAllester, D., Moser, M., Schmolze, J., Vilain, M. Research in Knowledge Representation for Natural Language Understanding - Annual Report, 1 September 1982 - 31 August 1983. Technical Report 5421, BBN Laboratories, Cambridge, MA, 1983.

[73] Sidner, C. L. Knowledge Representation for Natural Language and Planning Assistance: A Draft of Proposed Research. Bolt Beranek and Newman Inc., in preparation.

[74] Sidner, Candace L. "Plan parsing for intended response recognition in discourse." *Computational Intelligence 1* (1985), 1-10.

[75] Tversky, A. "Features of Similarity." *Psychological Review 84* (1977), 327-352.

[76] Vilain, Marc. KL-TWO, a Hybrid Knowledge Representation System. KRNL Group Working Paper, BBN Laboratories, 1984.

[77] Walker, Donald E.. *Understanding Spoken Language*. North-Holland, New York, 1978.

[78] Webber, Bonnie Lynn. *A Formal Approach to Discourse Anaphora*. Ph.D. Th., Harvard University, 1978. Also, Technical Report No. 3761, Bolt Beranek and Newman Inc.

[79] Weischedel, Ralph M. and Norman K. Sondheimer. "Meta-Rules as a Basis for Processing Ill-Formed Input." *American Journal of Computational Linguistics 9*, 3-4 (1983), 161-177.

[80] Benjamin Lee Whorf. *Language, Thought, and Reality*. The M.I.T. Press, 1956.

[81] Winograd, Terry. *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language*. Ph.D. Th., Massachusetts Institute of Technology, 1971. Also, Report No. TR-84, Project MAC, MIT.

[82] Winograd, Terry. A Procedural Model of Language Understanding. In *Computer Models of Thought and Language*, Roger C. Schank and Kenneth Mark Colby, Ed.,W. H. Freeman and Company, 1973, pp. 152-186.

[83] Winston, Patrick H. M.I.T. A.I. Progress Report. Massachusetts Institute of Technology, 1974.

[84] Woods, W.A. Semantics for a Question Answering System. Harvard University Computation Laboratory, September, 1967. Also, Ph.D. thesis, Division of Engineering and Applied Physics, Harvard University. Available from NTIS as PB-176-548, and reprinted with a new preface in 1979 by Garland Publishing, Inc. as a volume in the series: Outstanding Dissertations in the Computer Sciences.

[85] Woods, W.A. "Transition Network Grammars for Natural Language Analysis." *Communications of the ACM 13*, 10 (October 1970), 591-606.

[86] Woods, W.A., Kaplan, R.M. and Nash-Webber, B.L. The Lunar Sciences Natural Language Information System: Final Report. BBN Report 2378, Bolt Beranek and Newman Inc., Cambridge, MA, June, 1972.

[87] Woods, W.A. Semantics and Quantification in Natural Language Question Answering. In *Advances in Computers*, M. Yovits, Ed.,Academic Press, 1978, pp. 1-87.

[88] Woods, William A. Research in Natural Language Understanding: Quarterly Progress Report No. 6, 1 December 1978 to 28 February 1979. BBN Report 4181, Bolt Beranek and Newman Inc., Cambridge, MA, April, 1979.

Official Distribution List

Contract N00014-85-C-0079

	Copies
Scientific Officer Head, Information Sciences Division Office of Naval Research 800 North Quincy Street Arlington, VA 22217-5000 Attn: Dr. Alan L. Meyrowitz	1
Mr. Frank Skieber Defense Contract Administration Services Region - Boston 495 Summer Street Boston, MA 02210-2184	1
Director, Naval Research Laboratory Attn: Code 2627 Washington, DC 20375	1
Defense Technical Information Center Bldg. 5 Cameron Station Alexandria, VA 22314	12